

Nº7 – Mayo 2004

hardware

CONECTORES EAR Y MIC PARA EL +2A

input

ENTREVISTA A PEDRO GIMENO

CREANDO UNA AVENTURA CONVERSACIONAL CON Z88DK

al descubierto

LOS JUEGOS DEL ST2003, PATAS ARRIBA (Y II)

...y mucho más

www.speccy.org/magazinezx





Redacción:

Santiago Romero (SROMERO).
Federico Álvarez (FALVAREZ).
Pablo Suau (SIEW).

Ilustración de Portada:

Juanje Gómez (DEV).

Colaboraciones en este número:

Miguel G. Prada (MIGUEL).
Nestor Lucas (NÉSTOR LUCAS)

Maquetación en PDF

Álvaro Alea (ALEASOFT)

Contacto:

magazine@speccy.org



Redacción de MAGAZINE ZX

Bienvenidos a la séptima entrega de Magazine ZX. Se acerca el verano y con él más tiempo libre para disfrutar de nuestro Spectrum. Para amenizar la espera hasta las vacaciones os hacemos llegar este número calentito de nuestra revista. Veamos los contenidos de esta entrega:

Comenzamos con la parte software: En la sección Análisis descubriréis de la mano de MIGUEL dos juegos bastante conocidos. Por un lado el arcade "Harrier Attack" y, por el otro, el espectacular y colorido "Exolon". Por otra parte, nuestro colaborador NÉSTOR LUCAS pone "Al descubierto" más juegos del Speccy Tour 2003. Si en la anterior entrega ponía "patas arriba" una serie de juegos conocidos (entre ellos HATE y Tetris), este mes nos deleita con el análisis de juegos como CHASE HQ, Maziacs, Mikie, Jumping Jack o Arkadia.

En el apartado de Hardware descubrirás la manera de acoplar un lector de cassette externo al modelo de Spectrum +2A. Gracias a este montaje podrás salvar las limitaciones del tan criticado cassette original del +2A y utilizar tu datassette de toda la vida. En general podrás conectar cualquier fuente de audio externa, como un discman, o incluso cargar juegos desde el PC.

Los contenidos sobre programación este mes se centran en la programación en C con Z88DK. En esta entrega continuamos hablando sobre el completísimo compilador de C cruzado a través de un ejemplo basado en una aventura conversacional. De la mano de SIEW estudiaremos las estructuras necesarias para crear una aventura, y ejemplos de funciones para manipularlas.

En "Zona WWW" analizamos la Web de Digiexp: "El Spectrum Hoy". Imprescindible si quieres saber qué son y cómo se originaron las Ultracargas. La sección PanoramaZX nos acerca a las noticias más relevantes acaecidas estos últimos meses.

La parte de opinión de nuestra revista está encabezada por la entrevista a Pedro Gimeno, el autor del primer emulador español de Spectrum. Él trajo el Spectrum de vuelta a muchos de nuestros PCs cuando hubo que dar el salto de plataforma. Si quieres saber cosas sobre Pedro Gimeno y la gestación de este mítico emulador, no te pierdas la sección INPUT.

Y, por último, la sección de "Opinión" de MagazineZX, con una columna dedicada a los anuncios de compra-venta relacionados con el Spectrum. Una crítica directa con dosis de humor sobre ejemplos reales de anuncios.

Esperamos que os gusten los contenidos de esta entrega



Redacción de MAGAZINE ZX

Aunque el mundillo del Spectrum no es muy proclive a nuevas noticias y proyectos, en esta entrega tenemos muchas cosas que contaros. Desde el estreno internacional de la mejor tira cómica de Spectrum, hasta la celebración de los Aniversarios de importantes webs del mundillo. Desde MagazineZX os informamos sobre las noticias más destacables.

La tira de ECSS, en italiano

¿Alguien no conoce la famosa tira cómica de ECSS? Su autor, Pedrete, ha concedido permiso a Stefano Guida para la traducción de la tira cómica al italiano. El destino de la tira en Italiano será su publicación a partir del número 2 (Abril-Mayo) en el fanzine "Sinclair ZX Notizie", dentro de su web "Lo ZX Spectrum in Italia".

El próximo paso que debería plantearse la tira de Ecss sería su traducción al inglés, de forma que pueda ser disfrutada por los lectores de comp.sys.sinclair, o directamente de World Of Spectrum.

Chrome, clon del Spectrum

Con el paso del tiempo el hardware sufre averías (sobre todo en sus partes mecánicas), y llegará un momento en que será muy difícil encontrar Spectrums en perfecto estado. Podría decirse que el principal problema de nuestros Spectrums es el desconocimiento del esquema interno de la ULA, aunque no debemos olvidarnos de las endebles membranas de teclado, y la baja calidad de algunos modelos de moduladores de vídeo. También es un problema el hecho de que sólo es sencillo sacar vídeo RGB de los modelos 128K, +2 y +3. Ante todo esto, una de las prioridades de quienes pretenden preservar la plataforma es construir Spectrums desde cero. Clonar el Spectrum, pero utilizando componentes que permitan que todos puedan hacerlo, y hacerlo de una manera barata y rápida. Clonarlo, pero obteniendo la máxima compatibilidad con el software existente.

Chrome es un clon hardware del ordenador Spectrum. Como podéis ver en su web, tiene las siguientes características:

- Microprocesador Zilog Z80C, corriendo a 3.58 ó 7.1 Mhz.
- 64KB de ROM (BASIC del Spectrum) y 8K de ROM de disco.
- 160KB de memoria RAM (paginada en páginas de 16K) y 8KB de RAM de disco.
- Sonido de 3 canales, 7 octavas. Ruido blanco. Salida estereo. Altavoz interno de 1 canal.
- Interfaz de joystick compatible con Kempston.
- Puerto de Expansión.
- Interfaz I2C.
- Puerto paralelo de impresora.
- Posibilidad de conectar datacassette y disketera de 800KB de capacidad de disco.
- Gráficos de 256x192 píxeles a 16 colores.
- Salida RGB SCART.



Placa de Chrome conectada a teclado y disketera

Cabe destacar que el anuncio del prototipo indica que el diseño e implementación ya están finalizados, es decir, Chrome ya funciona al 100%.

Juegos de Spectrum en 256 colores

Y del hardware nos pasamos al software. Hace bastante tiempo que apareció el emulador Spec256, el cual permitía representar 256 colores en los juegos de Spectrum. Para poder usar estos 256 colores, los juegos debían ser adaptados por programadores, es decir, retocarlos para que utilicen "sets" gráficos diferentes. Debido al coste de tiempo y esfuerzo necesario para pasar los juegos (y a que no es algo que esté al alcance de cualquiera), apenas unos pocos juegos fueron pasados a 256 colores. Ahora ha aparecido otro emulador que los soporta (EmuZWin) y más gente se ha animado a pasar juegos a este formato "extendido" de color.



Abu Simbel Profanation en 256 colores

La aparición de EmuZWin parece bastante ligada a los grupos de usuarios ingleses de comp.sys.sinclair, donde hay una mayor base de programadores (en contraposición a Spec256, cuya base de usuarios era sobretodo hispanoparlante). Esto implica que más gente (con bastantes conocimientos técnicos) se ha interesado en el proyecto y es muy probable que el catálogo de juegos en 256 colores se amplíe aún más. La prueba de esto la tenemos en que en la misma página podemos descargar juegos de la Crap Games Compo de CSS en formato 256 colores, directamente adaptados por sus autores (es decir, programadores, y ahora con conocimientos sobre la adaptación a 256 colores). ¿Qué ocurriría ahora si alguno de estos programadores se centra en la adaptación de juegos a 256 colores para un emulador en lugar de hacerlo en la implementación de nuevos juegos para Spectrum?

Recordad que estos juegos especiales no pueden funcionar en el Spectrum real, tan sólo en estos dos emuladores que incorporan dicho modo de 256 colores. ¿Es esta una nueva manera de realizar remakes gráficos sin reprogramar código?

Aniversario doble



Sinclair QL, casualmente web de la Semana a finales de Abril en El Trastero

Este mes tenemos que celebrar 2 aniversarios. Por un lado, la completísima Web sobre el QL de Badaman "Sinclair QL. Recursos en Castellano" cumple 2 años informándonos sobre el QL y preservando todo tipo de información sobre este "desconocido" modelo de Sinclair. Este segundo aniversario coincide además con el 20 aniversario del QL (nacido en 1984).

Por otro lado, José Manuel anunciaba de esta forma en es.comp.sistemas.sinclair el tercer aniversario de la Web "El Trastero":

```
Autor: Jose Manuel
Asunto: Tercer aniversario del Trastero del Spectrum
Grupos de noticias: es.comp.sistemas.sinclair
Fecha: 2003-04-14 01:47:27 PST

Hola colegas, tal que así hace tres años comenzó la andadura
de esta página, en principio sólo dedicada a la búsqueda del
programa perdido y gracias a vuestras colaboraciones, se fue
ampliando con mas y nuevas secciones.

A la fecha hay mas de 170 MB, 1300 programas, y todo tipo
de material dedicado al Spectrum, y lo mejor, 37 mil visitas,
la mayoría de nuestro país, pero tambien desde todo el mundo.

Gracias, a todos, a los que colaboran, a los que nos leen, y en
especial a Santiago Romero por alojarnos en su excelente
servidor Speccy.org

Saludos, J.M.
```

Desde MagazineZX aprovechamos para felicitar a Badaman y a José Manuel por el excelente trabajo que realizando por el mundillo del Spectrum. ¡Felicidades!

Retornos y remodelaciones

Más sobre páginas Web: por un lado Santiago Romero remodela su web en Speccy.org para hacerla más legible y accesible, y permitir una más rápida actualización de sus contenidos. La web incluye ahora apartados (aunque vacíos por ahora) sobre otros sistemas retro. La parte de Spectrum es la más poblada por ahora, aunque carece todavía de contenidos nuevos.

Sin duda el retorno más interesante es el de Matranet. La entrega número 27 del fanzine más desenfa-dado del mun-do retro vuelve a nuestros na-vegadores. Especialmente ácida la crítica a Retrogames, revista retro re-cién aparecida y recién

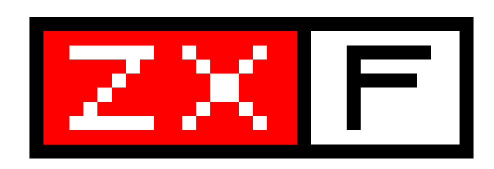
desaparecida (sólo ha durado 2 nú-meros), y en la que ha trabajado S.T.A.R.

Matranet es una revista diferente, con artículos retro y actuales, y centrada en la crítica realista en lugar de la habitual sistemática adulación de tópicos retro.

Nuevo disco de los AY Riders

En su página podéis escuchar el último disco de este grupo de artistas que se dedica a componer y hacer mixes de música de 8 bits en formato MP3. Van ya por su tercer álbum, titulado "8 BITS IS ENUFF?"

ZX Format nº 7



Logotipo de ZX Format

Aparece un nuevo número de "ZX Format: Spectrum Computing Today". Es la entrega número 7, correspondiente a la Primavera del 2004, y está repleto de contenidos sobre Spectrum.

The Complete Spectrum Rom Disassembly

Uno de los libros más valorados en el universo del Spectrum es sin duda "The Complete Spectrum ROM disassembly", donde se desgrana y desmenuza la ROM del Spectrum a través de un desemsamblado y comentado de la misma: todas las rutinas, variables, el intérprete de BASIC, etc. Es un libro imprescindible para comprender al máximo la arquitectura del Spectrum. Un libro imprescindible, y caro (y difícil de encontrar y conseguir).

Ahora un grupo de voluntarios lo ha escaneado y convertido a diferentes formatos, entre ellos texto plano y PDF, para que todos podamos disfrutar de sus contenidos. Sin duda este escaneo es un acontecimiento muy importante para todos los programadores interesados en el Spectrum.

Los usuarios de Spectrum españoles deberíamos tomar nota de este tipo de iniciativas, y realizar este tipo de labores. Tal y como hace José Manuel en El Trastero Del Spectrum, deberíamos dar una cierta prioridad (que en muchas webs está centrada en el software) a preservar los diferentes libros relacionados con el Spectrum. Obtener versiones online (en formatos texto, PDF y html) de los libros más importantes debería ser una obligación asumida por los propios usuarios y poseedores de los dichos libros. Es la mejor manera de compartirlos con todos los demás, y de que poder hojearlo electrónicamente preservando nuestro propio ejemplar (físico) del desgaste.

Éstas (y no son pocas) han sido las noticias más destacadas desde la anterior entrega de MagazineZX.

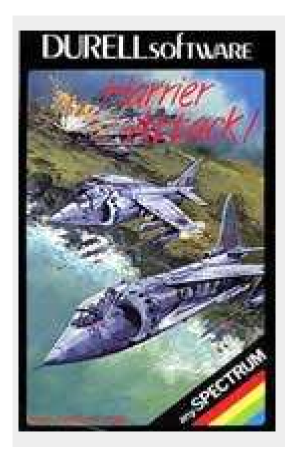
LINKS

- Tira cómica de ECSS (por Pedrete): http://www.speccy.org/pedrete/
- Fanzine Sinclair ZX Notizie: http://zxspectrum.hal.varese.it/
- Chrome, clon del Spectrum: http://aticatac.altervista.org/
- Juegos de Spectrum en 256 colores: http://www.arjun.150m.com/ZX256games.html
- Crap Games Compo (CSS): http://8bitorbust.info/cgc/
- EmuZWin: http://bonanzas.rinet.ru/e_downloads.htm
- Sinclair QL. Recursos en Castellano: http://www.speccy.org/QL/
- El Trastero del Spectrum: http://www.speccy.org/trastero/
- Retroweb de Santiago Romero: http://www.speccy.org/sromero/
- Matranet: http://www.matranet.net/
- AY RIDERS: http://ayriders.zxdemo.org/mp3.html
- ZXFormat issue 7: http://www.zxf.cjb.net/
- Escaneado de The Complete Spectrum ROM Dissasembly: http://freestuff.grok.co.uk/rom-dis/



Este mes contamos con los interesantes análisis de Harrier Attack y Exolon, ambos a cargo de MIGUEL. Como siempre, esperamos que los disfrutéis.

HARRIER ATTACK!



Título	Harrier Attack
Género	Arcade
Año	1983
Máquina	16K
Jugadores	1 Jugador
Compañía	Durell Software Ltd.
Autor	Mike A. Richardson
Otros comentarios	

- Crash Issue 1 (p. 48)
- Your Spectrum Issue 4 (p. 58)

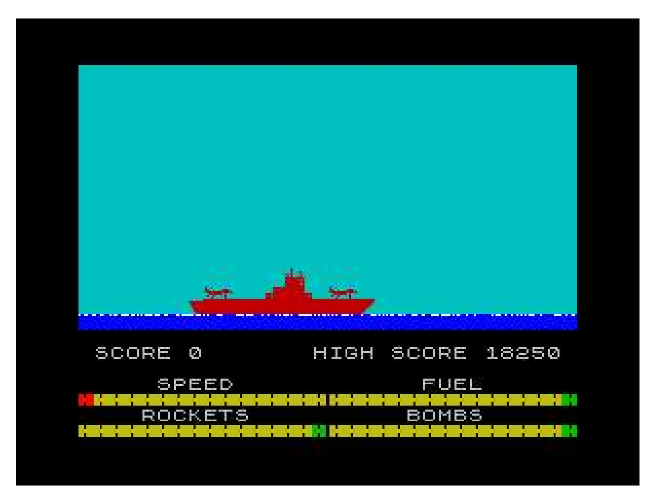
Hay momentos en que tenemos unos minutos libres y nos apetece jugar un rato, pero no lo hacemos ya que muchos juegos exigen que les prestemos atención durante mucho tiempo. Para esos quince minutos que tenemos sueltos sin nada que hacer Harrier Attack! puede ser una buena opción.

El juego que analizamos hoy es un Shoot'em'up con scroll horizontal en el cual manejamos un avión de combate Harrier, cuya característica más reseñada es la posibilidad de despegue vertical. Y así comenzamos el juego, despegando de un portaaviones en mitad del océano.



Pantalla de carga

El argumento del juego es inexistente, como los argumentos para justificar cualquier guerra, pero nuestra misión clara: internarnos en territorio enemigo y bombardear una ciudad con todo el arsenal de que dispone nuestro avión: misiles y bombas y no en poca cantidad. Para impedirlo el enemigo lanzará contra nosotros sus cazas, que si fallan el tiro intentarán colisionar con nosotros, cañones antiaéreos y tanques que nos dispararán en cuanto nos tengan a tiro, y os aseguro que empezaréis a ver la misión complicada en cuanto veáis el cielo repleto de disparos de los flaks. También encontraremos algún barco que nos lanzará misiles aunque, la verdad, no son muy difíciles de esquivar.



Comenzamos la misión

Como decíamos al principio, comenzamos nuestra misión en la cubierta de un portaaviones desde el cual despegamos, debemos tener cuidado de no dispararle ya que en ese caso al llegar al final de la misión no estará esperándonos para la recogida, con lo cual no podremos terminar el juego. Después de un recorrido sobrevolando el mar en el que nos encontraremos con

algún barco y aviones que intentarán derribarnos, nos adentramos tierra adentro a través de montañas y valles fuertemente defendidos por cañones antiaéreos y tanques. Esta es la zona más complicada del juego y habrá momentos en los que no sabremos por dónde nos vienen los disparos. Lo mejor que podemos hacer en esta zona es ir lo más rápido posible evitando los disparos desde tierra y los aviones enemigos y no entretenernos en disparar. A favor de esto está el complicado control de los disparos de los misiles, ya que cuando disparamos deberemos mantener el rumbo del avión en línea recta hacia el objetivo si queremos que el misil impacte en el blanco; si movemos el avión, el misil cambiará de trayectoria con él.



Un caza enemigo con malas intenciones

Es más sencillo usar las bombas para destruir los aviones enemigos situándonos encima de ellos y los misiles para los objetivos fijos en tierra. Al sobrepasar esta zona del juego, nos adentraremos en la ciudad, únicamente defendida por algún antiaéreo, y es aquí donde debemos aprovechar para hacer puntos descargando toda nuestra munición sobre las casas y vehículos que encontraremos, un vuelo raso y el machaqueo constante de la tecla de lanzar bombas nos ayudarán bastante. Después de esta fase poco nos queda que hacer, regresar a nuestra nave nodriza y aterrizar en ella.



Bombardeando civiles

La interfaz del juego es bastante completa, nos muestra la velocidad a la que vamos, que podemos incrementar y decrementar con las teclas de izquierda y derecha, las bombas y misiles que nos quedan y el combustible de que disponemos para terminar la misión. El nivel de combustible descenderá más rápido cuando vayamos a toda velocidad, pero aun así nos sobrará bastante para terminar la misión. Disponemos de una única vida, es decir, si destruyen nuestro avión nos enterraran con todos los honores o quién sabe si nos daran por desaparecidos en combate. Una vez que lleguemos al final de la misión el juego terminará y, si nuestra puntuación es digna de figurar entre las mejores, podremos introducir nuestro nombre en el hall of fame.

Los gráficos del juego cumplen con su cometido sin destacar en exceso, pero hay que tener en cuenta que este título data del año 1983 y los gráficos van acorde a lo que se diseñaba en esa época. El movimiento del juego es correcto y el scroll rápido pero se mueve a "saltos", ya que desplazan la pantalla de ocho en ocho pixels. El único problema apreciado con el movimiento es un ligero parpadeo en los barcos, por lo demás, correcto. Otro cantar suena con las opciones de configuración del juego ya que son nulas. Únicamente podemos seleccionar el nivel de dificultad de uno a cinco. La diferencia está en el número de enemigos que aparecen, la velocidad del juego, y la altura de las montañas: a mayor nivel de dificultad menos espacio para maniobrar a más velocidad y esquivando más disparos, pero esto puede ser realmente divertido. Nuestro avión responde a las teclas de control a la perfección, aunque tendremos que adaptarnos a los cursores para manejarlo más el nueve para arrojar las bombas y el cero para lanzar misiles.

El sonido se limita a unos cuantos efectos de sonido al destruir los objetivos. Nada de música.



Finalizamos la misión

Harrier Attack! es un juego entretenido y correcto tecnicamente, más teniendo en cuenta que fue programado para cualquier máquina Sinclair a partir de los 16KB de memoria. Nos permitirá pasar unos minutos picados pero sin pretender que esos minutos se extiendan demasiado, como puede pasar en juegos similares (me viene a la cabeza el Penetrator, por ejemplo). Baste decir que pasar el juego completo en nivel intermedio lleva unos dos minutos y medio sin tener mucha práctica.

Valoraciones		
Originalidad:	[6]	
Gráficos:	[6]	
Sonido:	[5]	
Jugabilidad:	[8]	
Adicción:	[8]	
Dificultad:	[6]	

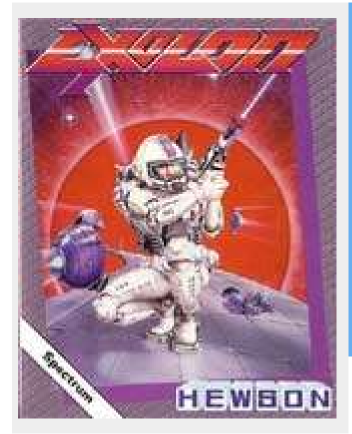
Trucos:

Puedes encontrarlos en The Tip Shop http://www.the-tipshop.co.uk/cgi-bin/search.pl?name=Harrier+Attack

Descárgalo de:

- WOS http://www.worldofspectrum.org/infoseek.cgi?regexp=Harrier+Attack

EXOLON



Título	Exolon
Género	Arcade
Año	1987
Máquina	48K
Jugadores	1 Jugador
Compañía	Hewson Consultants
Autor	Raffaele Cecco
Otros comentarios	

- Crash Issue 43
- Your Sinclair Issue 20 (p. 35)

Lo habitual en esta sección es comentar juegos que no son demasiado conocidos, aunque no me he podido resistir a la tentación de analizar un gran juego que me ha tenido absorbido los últimos días: Exolon, de Raffaele Cecco.

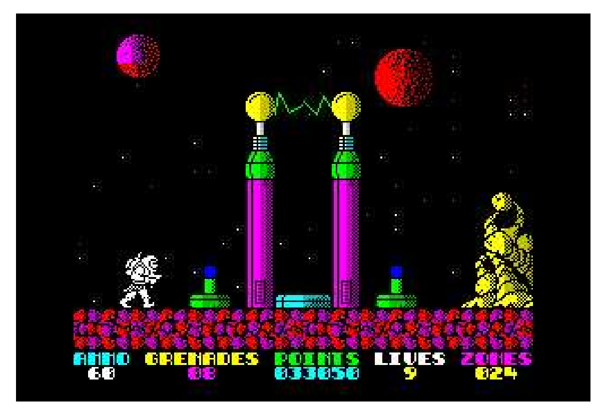


La pantalla de carga

Poca presentación necesita Cecco. Como autor de juegos ha creado verdaderas maravillas de la programación en las que llevaba la máquina a sus límites. Títulos como Cybernoid, Solomon's Key, Stormlord o este Exolon que voy a diseccionar hoy hablan por si mismos.

Nuestra misión es limpiar un desconocido planeta de los alienígenas que lo habitan, que no son pocos. Para ello, estamos equipados con un lanzagranadas y un fusil para disparar a todo lo que se mueva... y a lo que no también. La capacidad que tenemos para llevar munición es limitada, aunque vamos a ir encontrando más según avancemos en el juego y en ningún momento notaremos su escasez.

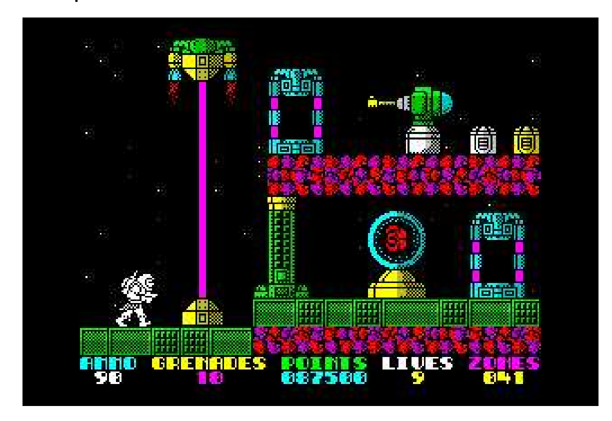
El juego es un plataformas de pantalla estática, es decir, sin scroll, en el cual nos iremos topando con un desafío diferente según avanzamos, y necesitaremos plantear una estrategia determinada para superar cada pantalla, no debiendo limitarnos a disparar sin ton ni son. Suele haber dos caminos diferentes para pasar cada zona y, según la que decidamos, normalmente una opción más complicada o afrontando más enemigos que la otra, así nos puntuaran al final de cada una de las cinco fases de las que está compuesto el juego. A más peligros afrontados, más puntos.



Finalizando el nivel uno

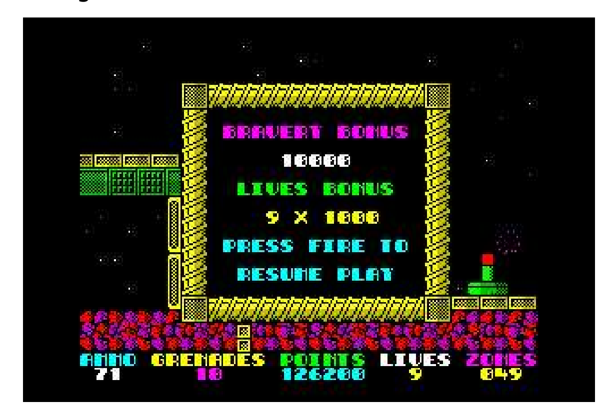
En cada una de las 25 pantallas en que se dividen las cinco fases encontraremos enemigos que se mueven con unos patrones predeterminados, y que no son muy complicados de eliminar estudiando sus movimientos. También encontraremos cañones fijos que nos dispararan al detectar que estamos en su línea de tiro. De estos cañones, algunos conseguiremos destruirlos con las granadas y otros solamente podremos superarlos y dejarán de disparar cuando estemos muy cerca de ellos, aunque a nuestro favor tenemos que los proyectiles de estos últimos

podemos destruirlos con nuestros disparos. Uno de los obstáculos más complicados son unas plataformas que salen del suelo con una cadencia aleatoria y, si nos pillan, vida menos en la cuenta.



¿Complicado? y lo que nos queda...

Hay zonas que encadenan varias de estas plataformas que son auténticas pesadillas para sobrepasarlas. Otro obstáculo bastante complicado, más por la sorpresa que nos causa su disparo nada más entrar en la pantalla que por la dificultad en destruirlo, es un misil controlado por una especie de radar que se dirigirá hacia nosotros y es muy complicado de esquivar. Para destruirlos, la única manera es lanzar una granada contra su control o radar.

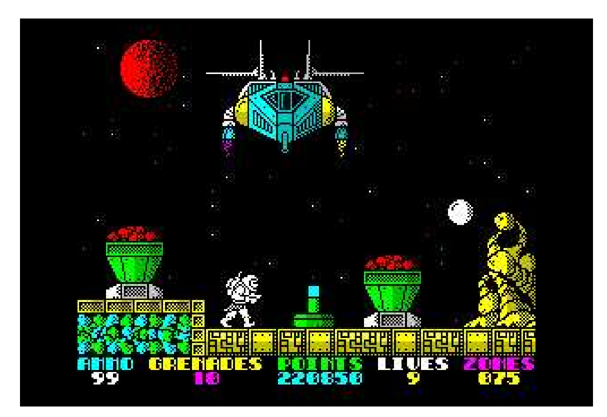


Momentos de calma y satisfacción

Técnicamente el juego roza un nivel muy alto. Los movimientos son suaves, y el teclado redefinible a nuestro gusto. También contamos con la opción de controlar el juego mediante un joystick de norma Kempston o Sinclair Interface 2. En cualquier caso, el protagonista responde a nuestros requerimientos sin demora. Si se le puede poner una pega en los movimientos es una apreciable ralentización cuando hay muchos sprites simultáneamente en pantalla, especialmente cuando liberamos una pequeñas bolitas rojas reunidas en una esfera transparente. Pero hablamos de más de 20 sprites moviéndose sin pausa, de un tamaño más que considerable y a todo color. Aun así el juego es controlable y jugable totalmente. ¿Se debe quejar uno por esto? Yo creo que no.

Los gráficos son coloridos, pero no coloridos sólo por resaltar. Los colores están aplicados con corrección y un gusto sobresaliente, y con un diseño excelente, de buen tamaño y que fusionan muy bien unos con otros. Las 125 pantallas son un derroche de imaginación y diseño, da gusto moverse por ellas.

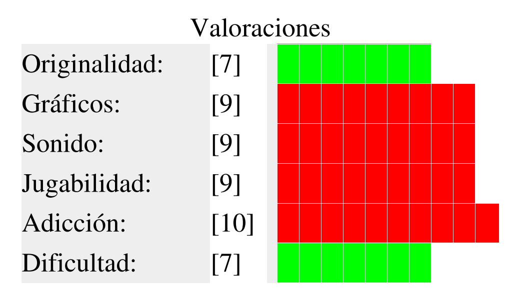
El sonido merece una mención especial, aunque hay que diferenciar entre cargar el juego en una máquina de 48KB o en una que disponga del chip de sonido que llevan los modelos de 128KB. Simplemente no hay color. Comenzando por la melodía del menú, una pequeña obra de arte que se muestra en todo su esplendor con el chip AY funcionando y pasando por los efectos de sonido al lanzar una granada, disparado o recogiendo las diferentes cargas de munición que encontraremos a lo largo de nuestro recorrido. No vamos a notar un despilfarro de sonido ni una música machacona según jugamos, que no la hay, pero cada vez que el Spectrum suena lo hará como esperamos.



Bonitos gráficos y no nos atacan... de momento

Como os habréis dado cuenta el juego me gusta y no trato de ocultarlo, es uno de esos imprescindibles que no puede dejarse de jugar y que una vez que te pones con él no puedes soltarlo hasta que lo terminas, cosa complicada ya que, aunque no es de los juegos más difíciles y lleva una curva de aprendizaje bastante suave, hay momentos en que puede llegar a desesperarnos un poco. Dejad de jugar al Pro Evolution Soccer un rato y veréis que no es necesario un juego con un trillón de polígonos para pasar un rato lúdico excelente.

Valoraciones



Originalidad:	[7]
Gráficos:	[9]
Sonido:	[9]
Jugabilidad:	[9]
Adicción:	[10]
Dificultad:	[7]

Trucos:

Puedes encontrarlos en The Tip Shop http://www.the-tipshop.co.uk/cgi-bin/search.pl?name=Exolon

Descárgalo de:

- WOS http://www.worldofspectrum.org/infoseek.cgi?regexp=^Exolon$&pub=^Hewson+Consultants+Ltd$

MIGUEL



FALVAREZ

CONECTORES EAR Y MIC PARA EL +2A.

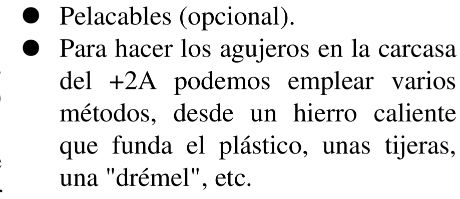
Uno de los cambios más significativos que se introdujeron cuando Amstrad se hizo cargo de la producción de ordenadores Sinclair fue la inclusión de un lector de cassette acoplado al ordenador. De esta forma, el usuario dejaba de depender de un lector externo y (teóricamente), aumentaba la fiabilidad al leer y grabar las cintas.

No obstante, la experiencia nos dice muchos años después que dicha fiabilidad no era tal. Bien porque el lector del +2A esté en plan rebelde y no lea las cintas, bien porque con el paso del tiempo esté estropeado y tenga mal arreglo, siempre puede ser interesante volver a disponer de la opción de cargar y grabar software en un lector externo.

No se trata de un montaje difícil. Eso sí, vamos a tener que modificar (levemente) la apariencia externa del ordenador. Quiero agradecer a MIGUEL que me prestara su hospitalidad y su ordenador para hacerle unas fotos al montaje.

MATERIAL NECESARIO

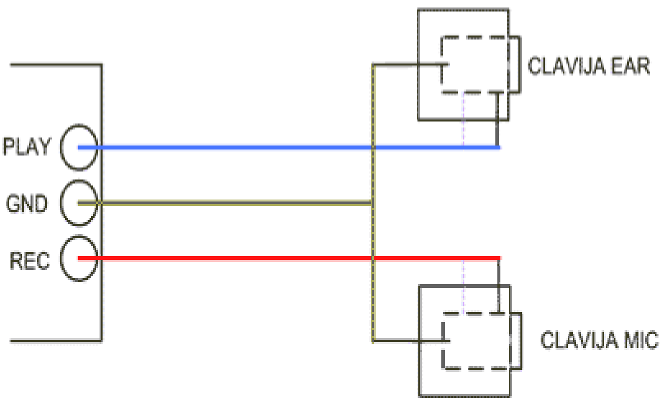
- 2 clavijas mini jack mono para montaje en chasis (si son estéreo también valen, en nuestro caso teníamos un par a mano y las utilizamos en vez de comprar otras nuevas).
- Un destornillador de estrella (para abrir el +2A y extraer las partes necesarias).
- Cable de un hilo conductor (estimo que con un metro tendremos de sobra).
- Soldador y estaño.
- Pelacables (opcional).
- Para hacer los agujeros en la carcasa del +2A podemos emplear varios métodos, desde un hierro caliente que funda el plástico, unas tijeras, una "drémel", etc.



IDENTIFICACIÓN DEL OBJETIVO

Abrimos el ordenador retirando los tornillos inferiores y los dejamos a buen recaudo. Como siempre que se abre un +2A, hay que tener cuidado con varios elementos que unen la carcasa superior a la inferior. Dos de ellos son los frágiles conectores que unen la membrana del teclado a la placa base. El otro es un cable de cuatro hilos (ver Imagen 1) que lleva la señal del lector de cintas a la placa, y que será objeto de nuestro estudio, ya que nos dará la clave para conectar las dos clavijas que pretendemos añadir.





Esquema 1. Conexionado



Imagen 1. Cable de datos del cassette

Numeramos los colores de derecha a izquierda, así obtenemos la siguiente correspondencia (ignoro si el el orden del color de los cables será el mismo en todos los casos):

1. Cable marrón.
2. Cable azul.
3. Cable rojo.
4. Cable blanco.

Estos cables van a la placa del lector de cassettes. Sólo necesitamos, pues, una pista que nos indique cuál es la señal que transita por cada uno de ellos. Y esa pista la encontramos serigrafiada en la placa base (ver Imagen 2).

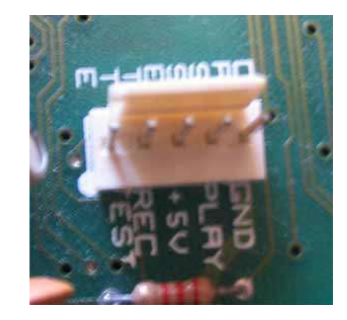


Imagen 2. Conector del cassette en la placa base

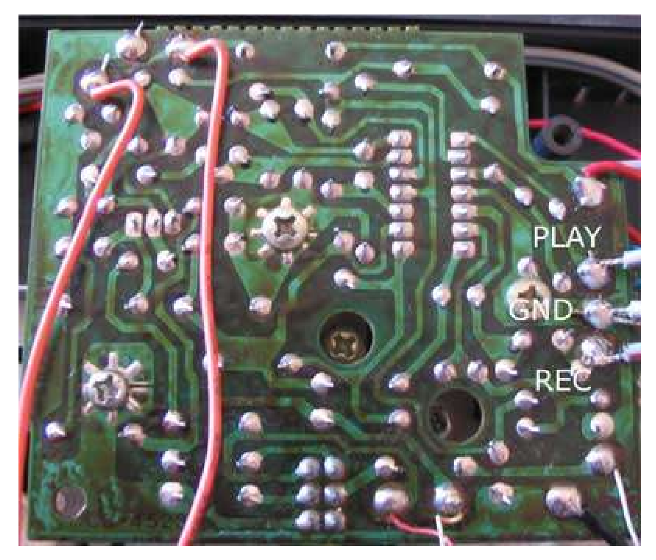


Imagen 3. Placa del lector de cassette

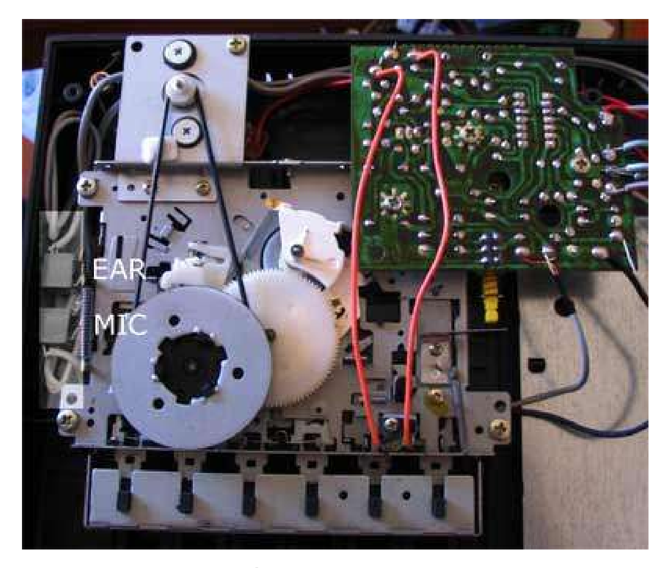


Imagen 4. Ubicación de las nuevas clavijas EAR y MIC

Por tanto, llegados a este punto, ya podemos establecer la correspondencia que andábamos buscando.

Color	Señal
Marrón	GND (masa)
Azul	Play
Rojo	+5V
Blanco	Rec

De estas señales, la señal de alimentación no nos interesa. Seguimos los colores de las otras tres para localizar los puntos de soldadura, que están indicados en la Imagen 3.

Por tanto, ya está claro de dónde sacaremos cada señal, pero queda reflejado aún más claramente en el Esquema 1. La toma de masa irá a ambas clavijas, y las dos de señal, una a cada una de ellas. En el caso de que las clavijas de las que dispongáis sean estéreo (en el esquema queremos reflejar este particular representando el segundo pin en color morado y línea de trazo discontinuo), se deberán soldar ambos pines de señal.

A continuación lo mejor es soldar las clavijas antes de montarlas en el chasis. Las clavijas van a ir situadas en la carcasa superior en el lateral derecho (en la Imagen 4 podemos ver la ubicación en una fotografía con el montaje ya terminado). Tomamos medida de los cables para que no se queden tirantes y soldamos según el esquema.

El siguiente paso consiste en agujerear la carcasa. Ya que la modificación va a resultar definitiva, es recomentable tomar bien las medidas de la ubicación, y hacer los agujeros con cuidado. Si disponemos de un minitaladro, resulta ideal para este tipo de tareas, pero, si no es así, podemos llegar a realizar las incisiones con una herramienta tan tosca para este menester como pueden ser unas tijeras (manejadas con destreza pueden obrar maravillas). Desenroscaremos la arandela exterior de las clavijas y las colocaremos en su lugar correspondiente, para después dejar-las firmemente sujetas a la carcasa plástica enroscando de nuevo dicha arandela, como se puede apreciar en la Imagen 5.



Imagen 5. Detalle de las clavijas ya colocadas

Ya sólo nos resta volver a conectar todo lo que hemos ido desmontando y cerrar el ordenador, teniendo precaución de no pillar ninguno de los cables que hemos añadido. El nuevo aspecto externo del ordenador se muestra en la Imagen 6.

El significado de las clavijas es el mismo que en la gama Sinclair carente de cassette incorporado, esto es, EAR es la entrada de datos desde la cinta y MIC es la salida hacia la cinta. Por último debemos regular el azimut, si procede, y disfrutar de nuevo de nuestro datassette de toda la vida.



Imagen 6. Detalle de las clavijas ya colocadas (exterior)



NÉSTOR LUCAS

LOS JUEGOS DEL ST2003, PATAS ARRIBA (y II)

Esta es la segunda y última entrega de la serie de artículos en los que NÉSTOR LUCAS, flamante ganador del ST2003, destripa todos y cada uno de los juegos de dicha competición.



Título	Chase H.Q.
Género	Arcade/Action
Año	1989
Máquina	ZX Spectrum 48K/128K
Jugadores	1 Jugador
Compañía	Ocean Software Ltd
Autor	John O'Brien, Bill Harbison & Jonathan Dunn
Categoría en el Tour	Racing

En toda competición de videojuegos que se precie no puede faltar un juego de carreras, que en esta ocasión añade el aliciente de basarse en las típicas persecuciones automovilísticas de las series policíacas.

Al principio de cada carrera, recibimos un comunicado de la central indicándonos los detalles del vehículo en el que un fugitivo está dándose a la huida. Nuestra misión es alcanzarlo y detenerlo.

Para ayudarnos en la captura del sospechoso, el coche dispone de tres turbos, que conviene guardar hasta que lo tengamos a la vista, pues la forma de detener al fugitivo es un tanto drástica: golpeando su coche hasta dejarlo inutilizable.

El juego consta de cinco circuitos, en los que perseguiremos sucesivamente a un sospechoso de asesinato en primer grado, a otro de robo a mano armada, a un vendedor de drogas, a un presunto secuestrador y a un sospechos de espionaje y asesinato. Todo ello protagonizado por una pareja de policías de la que nadie duda de su parecido con los protagonistas de la serie (de la misma época que el videojuego) titulada 'Miame vice'.

CHQ-01. Desarrollo del juego.

Como en todo juego de carreras que se precie, en Chase H.Q. tenemos un checkpoint que alcanzar antes de que se agote el tiempo para continuar con el juego. Siguiendo con el argumento, este checkpoint no es otro que avistar al sospechoso que ha iniciado la huída.

Antes de alcanzar al fugitivo, una de las avenidas por las que circulamos se bifurcará, teniendo que elegir uno de los dos caminos en los que se divide. Si bien por cualquiera de los dos caminos llegaremos al mismo punto y podremos atrapar al malo, recibiremos información en forma de flecha o comunicado por radio indicando cuál ha sido el camino tomado por el sospechoso, es decir, cual será nuestra ruta más corta.



Hay que escoger bien el camino en la bifurcación

Dado que el juego se centra en una persecución por las calles de una ciudad, tendremos que prestar especial atención al resto de coches que circulan por la carretera, evitando en todo lo posible tener incidentes con ellos, pues aparte de que queda muy mal y da mala imagen a las fuerzas del orden, nos

resta velocidad, con lo que supone eso para el tiempo restante para alcanzar al sospechoso y, además, se resetea el valor de los puntos de bonificación que se nos otorgan por cada coche no implicado en la huída que adelantemos sin causarle daños.



En la tercera fase, en lugar de túneles pasaremos por debajo de unos arcos

Por otra parte, en nuestra persecución también atravesaremos túneles donde supuestamente el roce con las paredes debería frenarnos, como sucede en la recreativa original. Sin embargo, algún error de programación nos permite, de manera bastante aconsejable por cierto, pegarnos a la pared una vez dentro del túnel, para así esquivar al resto de coches que circulan por la vía. No sucede lo mismo con los arcos de la tercera persecución, que estos si hacen polvo a nuestro coche, llegando a pararnos por completo ante cualquier impacto.

Otros elementos de la carretera, como las vallas de obras y los restos de las mismas, supuestamente deberían frenarnos pero, al igual que con las paredes del túnel, los programadores debieron pensar que había que darle alguna ventaja a los sufridos jugadores de la versión para Spectrum, y así resulta que nos los podemos llevar por delante sin sufrir perjuicio alguno.

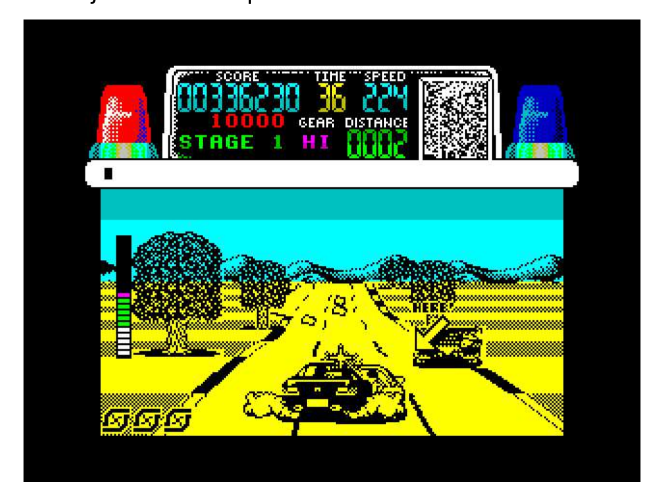
Una vez avistemos el coche sospechoso, iniciaremos la persecución propiamente dicha. Se encenderán la sirena y los rotativos de nuestro vehículo, y a base de sucesivos golpes tendremos que detener la huida del fugitivo.

Hay varias formas de hacerlo, con diversos trucos divulgados por Internet. Uno de ellos, por ejemplo, nos indica que justo antes de golpear al coche del sospechoso, reduzcamos la marcha del vehículo e instantáneamente volvamos a cambiarla, para así, si lo hacemos bien, obtener dos puntos de golpe en lugar de solo uno.

Otro truco que a mí personalmente me ha resultado bastante bien es, en lugar de golpear al coche de la huida desde atrás, tratar de adelantar y golpearle lateralmente. Bien hecho resulta igualmente en dos puntos de golpe en lugar de sólo uno, con la ventaja adicional de que apenas perdemos velocidad.

Sin duda el modo más espectacular de puntuar alto en los golpes contra el coche huido es aprovechar los desniveles de la carretera, para cuando haya un salto caer encima de él, puntuando de nuevo por dos golpes en lugar de por uno solo.

Y no podemos olvidar los turbo que mencionábamos al principio del comentario de este juego. En tramos amplios, rectos, y curvos también aunque con más cuidado, podemos aprovechar los turbos para causar varios daños al vehículo en huida, siendo esta quizás la mejor forma de provocar su detención.



Hay varias formas de golpear al fugitivo y provocar su detención

CHQ-02. La misteriosa puntuación extra y la expliación de Russell Marks

Durante la competición, varios jugadores detectamos un curioso comportamiento en el sistema de puntuación del juego. Resulta que en determinadas circunstancias, al finalizar cada fase aparecía una puntuación extra de 1.280.000 puntos sin que se conociera el origen de esta inusitada cifra.

Como suele ocurrir en estos casos, la situación se comentó varias veces en los grupos es.comp.sistemas.sinclair y comp.sys.sinclair, dándose posibles explicaciones de lo más variopintas. Hasta que un buen día, uno de los participantes del Tour, habitual de c.s.s., y gran partícipe del movimiento Speccy en Internet nos sorprendió con el descubrimiento de un bug en el código del juego, que era el causante de esta misteriosa bonificación.

Comenzaba Russell Marks su explicación con un desensamblado del código máquina del juego, descubriendo en la dirección 8B3Dh lo siguiente:

```
        EX AF,AF'
        LD B,A
```

En esta posición, el código calcula la puntuación de bonificación correspondiente al dígito de menor peso del temporizador, multiplicando 5000 por el valor contenido en B mediante un bucle posterior con el tradicional DJNZ, siendo el valor de B el que se carga con la instrucción LD B,A que vemos en el segmento de código.

Sin embargo, antes de cargar B con el valor del acumulador, se realiza un intercambio de registros,

con lo que en lugar de obtener el valor de la cifra de menor peso del temporizador que estaba almacenado en A, se carga con el valor de A', sea éste cual sea.

Así pues, ¿qué contiene A' en el momento de realizar este cálculo?

Continuaba Russell su explicación en c.s.s. diciendo que existen tres posibilidades:

- Si la cifra de menor peso del temporizador es cero, la parte del código que contiene el error en el cálculo de la bonificación no se ejecuta, pues no hay que calcular el valor correspondiente a la cifra de menor peso, y por tanto, no hay posibilidad de obtener la puntuación extra.
- Si la cifra de mayor peso del temporizador es cero, entonces A' no se carga con ningún valor en ninguna parte del código, por lo que el resultado de la bonificación extra es ciertamente aleatorio.
- Si ninguna de las cifras es cero, se obtiene como puntuación extra el resultado de multiplicar 5000 por el valor BCD más alto de nuestra puntuación, salvo cuando esta es menor de un millón, situación en la que el bucle se ejecuta 256 veces, obteniendo una bonificación de 1280000 puntos.

En otras palabras, todo lo que hay que hacer para conseguir la puntuación extra es:

- Asegurarse de terminar siempre con la cifra más baja del temporizador siendo distinta de cero.
- En las fases posteriores a la primera, finalizar cada carrera dejando menos de diez segundos en el temporizador, y dejar a la suerte el valor de la puntuación extra obtenida.

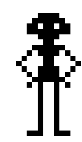
De esta forma queda explicado el misterio de la misteriosa bonificación.



Título	Maziacs
Género	Arcade/Maze
Año	1983
Máquina	ZX Spectrum 48K/128K
Jugadores	1 Jugador
Compañía	DK'Tronics Ltd
Autor	Don Priestley
Categoría en el Tour	Maze

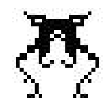
Estamos ante un juego que a jucio personal no era de los más adecuados para el Tour. Y no precisamente porque le faltara calidad, si no más bien por lo aleatorio de los resultados. Pero veamos primero un análisis algo más detallado del juego.

MAZ-01. Los elementos del juego



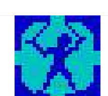
El protagonista

Pues bien, éste es nuestro protagonista, el caballero que arriesga su vida en el laberinto a la búsqueda del tesoro.



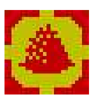
El Malo

Este bicho con ese aspecto monstruoso es el malo que ronda por el laberinto. Debemos evitarlo a toda costa, salvo que vayamos armados con la espada. En ocasiones no queda más remedio que enfrentarse a él desarmado, y con un poco de suerte más que de habilidad podremos salir airosos del enfrentamiento, aunque la norma es más bien todo lo contrario.



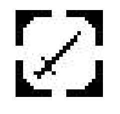
Los prisioneros

El laberinto esta ocupado además por mazmorras en las que nos encontraremos prisioneros que, gentiles ellos, si les preguntamos nos mostrarán el camino hacia el tesoro, o si ya lo tenemos en nuestro poder, hacia la salida. Algunos no se explican cómo conociendo tan bien el laberinto siguen aun prisioneros, algo que no sucede en el nivel cuatro de juego, donde para indicarnos el camino tenemos que liberarles, vamos, que desaparecen del juego.



El aporte nutricional

En la parte derecha de la pantalla vemos una barra de energía que por momentos va decreciendo. Esa es la energía de que dispone nuestro caballero en su búsqueda del tesoro, y que por todos los medios tenemos que evitar que llegue a cero. Para eso se encuentran repartidos por el laberinto varios bloques de alimento como el ilustrado. A la hora de cogerlos, tengamos en cuenta que también nos hará falta energía en el camino de regreso con el tesoro.



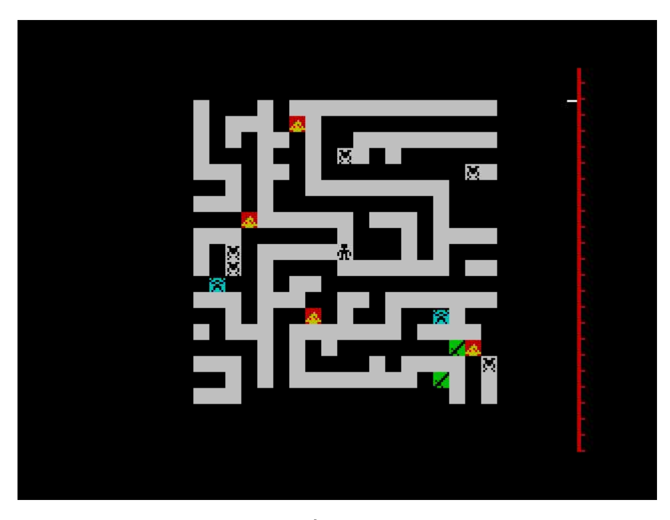
La espada

Es la única arma de que disponemos, con la peculiaridad que cada vez que nos enfrentamos a un bicho, perdemos la espada. Así pues, escondidas por el laberinto hay también repartidas espadas que tendremos que recoger cada vez que usemos una en despejar el camino de bichos.



El tesoro

Nuestro objetivo final es conseguir este tesoro y volver al punto de partida con él. Cuando cargamos con el tesoro no podemos llevar espada, así que más nos vale haber despejado antes el camino de regreso de bichos.



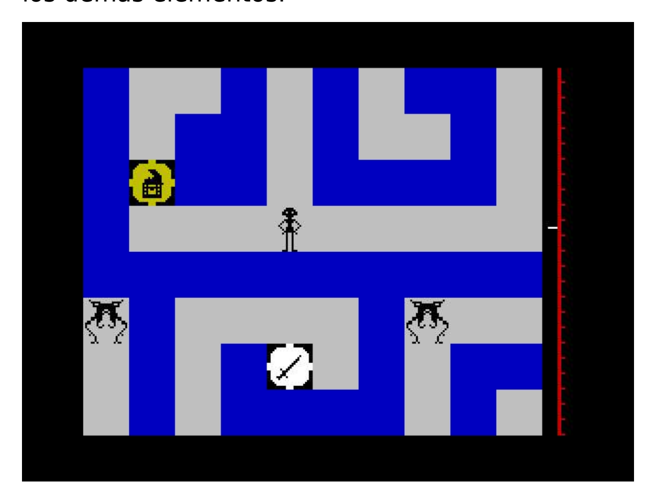
El mapa

En cualquier momento del juego podemos consultar el mapa del laberinto para orientarnos y decidir qué camino seguir. Hay que tener presente que el hecho de inspeccionar el mapa no para el juego, así que si hay algún bicho cerca, corremos el riesgo de ser atacados por sorpresa, lo que significa en la mayoría de los casos que aunque llevemos la espada el bicho podrá con nosotros.

MAZ-02. El juego

Como ya he comentado al principio, el juego es en sí un laberinto, cuyo sistema de puntuación se basa en la diferencia entre el camino óptimo entre el punto de partida y el tesoro (ida y vuelta), y el camino seguido por nosotros, realizando una sencilla regla de tres en la que se otorgan cien puntos al camino óptimo.

A esta puntuación hay que añadirle un plus en función del nivel escogido, pues cada aumento de nivel significa un incremento de la dificultad, tanto en el tamaño del laberinto, como en el comportamiento de los demás elementos.



Encontrar el tesoro es nuestro principal objetivo

Así, cuanto más alto sea el nivel (de 1 a 4), más inteligentes serán los bichos, es decir, que en el nivel 1 su movimiento es más bien errático, siendo en el nivel 4 dotados de cierta inteligencia que les hace capaces de perseguirnos y encontrar caminos hacia donde nos encontramos.

Otro cambio a tener en cuenta en los niveles superiores es el comportamiento de los prisioneros. En los niveles más bajos les podemos preguntar por el camino hacia el tesoro tantas veces como nos apetezca. Sin embargo, en el nivel más alto nos conviene preguntar sólo cuando es necesario, pues al preguntarles les liberamos, y ya que se conocen tan bien el laberinto, pues se se van para nunca volver.

Pocos consejos se pueden dar sobre este juego, en el que el azar cobra un importante papel. Y es que más que de nuestra habilidad como jugadores, la puntuación final depende de la suerte que tengamos en la generación del laberinto. Todo lo más que os puedo decir es que no despreciéis una buena consulta al mapa de vez en cuando.



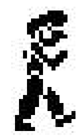
Título	Mikie
Género	Arcade/Action
Año	1985
Máquina	ZX Spectrum 48K
Jugadores	1 Jugador
Compañía	Imagine Software Ltd
Autor	Jonathan Smith & Martin Galway
Categoría en el Tour	Arcade

Sin duda uno de los juegos más populares del Tour, junto con el Tetris, el Supertest y el Chase H.Q., es Mikie. Y no es para menos con un arcade al más puro estilo tradicional.

MIK-01. La base del juego

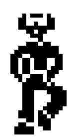
El juego lo protagoniza un estudiante de instituto, cuyo nombre no sorprende, Mikie. Su objetivo es enamorar a una chica que nos espera fuera de las dependencias del instituto y, para lograrlo, debe recoger todos los corazones que hay repartidos por las diferentes dependencias del centro. Un argumento bastante similar al también arcade Popeye, todo hay que decirlo, pero con un toque roquero que se pone de manifiesto en la pantalla del menú con la canción "A Hard Day's Night" de The Beatles.

MIK-01.1. Los personajes del juego



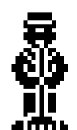
Mikie

Éste es Mikie, el adolescente protagonista del juego.



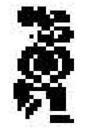
El profesor

El profesor tratará de poner fin a la huída de clase emprendida por Mikie. Para ello nos perseguirá por todo el instituto, aumentando su velocidad cuanto más tiempo permanecemos en una estancia, y siendo además el único que cuando se cabrea de verdad nos ataca tirándos algo que parece una boca.



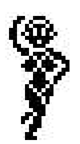
El conserje

Nos lo vamos a encontrar en todos los pasillos y en algunas estancias del instituto, llegando hasta triplicarse en el patio donde nos espera nuestra enamorada. Eso sí, un gran trabajador, porque allá donde nos persiga, estará siempre barriendo.



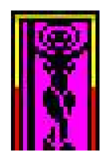
El cocinero

En el gimnasio y en la cocina aparece este otro personaje del personal del centro. Es como el profesor, que nos persigue para atraparnos y darnos una reprimenda. Pero se diferencia de éste en que su velocidad no aumenta con el paso del tiempo, y que no nos lanza nada, con lo que resulta más inofensivo.



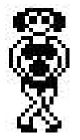
Las bailarinas

Están en el gimnasio, con su clase de baile. Van todas en formación hasta que una de ellas improvisa un nuevo paso de baile y se sale de la formación. Hay que esquivarlas a toda costa, pues aunque no detienen a Mikie, como adolescente que es le vuelven loco las chicas, y se queda atontado unos instantes. Los justos para que le atrape el profesor.



La chica del cambiador

Detrás de las puertas sorpresa de los pasillos se puede esconder esta sugerente chica, que igual que eleva la moral de nuestro personaje, hace lo propio con los puntos, dándonos una bonificación extra. Solo hay una chica por pasillo.



La novia de Mikie

Una vez recogidos todos los corazones nos espera esta chica, la chica de los sueños de Mikie, en el jardín.

MIK-02. Jugando a Mikie



La clase

Empezamos el juego en la clase de Mikie. Su novia le espera fuera, en el jardín, y para demostrarle su amor, nuestro playboy protagonista debe recoger todos los corazones desperdigados por el instituto.

En la clase los corazones se encuentran escondidos debajo de los pupitres. Para poderlos coger tenemos

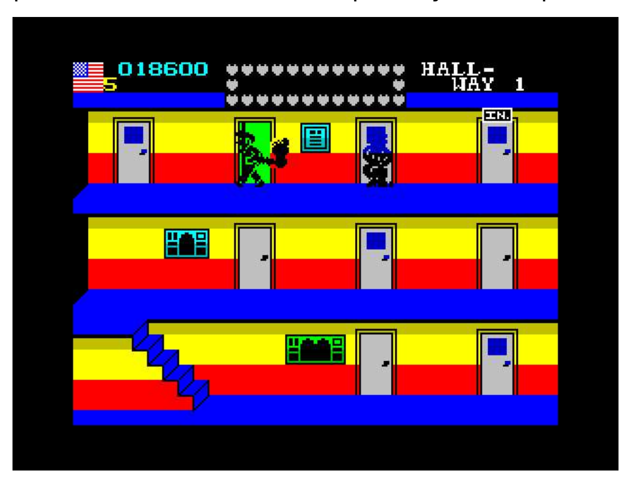
que despejar primero el pupitre correspondiente y, como los compañeros de clase son tan poco colaboradores, sólo hay una opción: empujarles.

Como vemos durante el juego, los corazones aparecen siempre de color negro, salvo por el hecho de que secuencialmente hay uno que está parpadeando. Si recogemos los corazones según parpadeen, obtendremos una puntuación más alta por el corazón recogido de esta forma. Además, si logramos que todos los corazones sean recogidos al parpadear haremos un nivel perfecto, lo que supone una bonificación extra.

Además de la puntuación extra por recoger los corazones en el momento adecuado, cada estancia del juego oculta una bonificación extra que iremos desvelando según avancemos por las diferentes instalaciones del centro.

En el caso de la clase donde se inicia el juego, tenemos que situarnos frente a la mesa del profesor, mirando hacia la pizarra. Allí gritamos tres veces y veremos que aparece un objeto sobre la mesa, señal de que hemos obtenido 1000 puntos extra. A modo de indicación para esta primera ocasión, en la imagen que da pie a este apartado podéis ver a nuestro amigo Mikie gritando en la ubicación indicada.

Cuando hayamos recogido todos los corazones, con o sin bonus, se iluminará un cartel en una de las puertas indicándonos la salida. Solo queda ir a esa puerta, abrirla (botón de disparo), y salir al pasillo.



El primer pasillo

Los pasillos son las vías de comunicación entre las demás estancias del instituto, como es de esperar. La siguiente habitación a la que debemos dirigirnos tiene un cartelito iluminado sobre la puerta, con el texto 'IN'. Se entra igual que salimos de la clase.

Además de la puerta que nos llevará a la siguiente clase, en los pasillos hay más puertas que se pueden abrir, que son las que no tienen ventana. Estas puertas esconden sorpresas. En la mayoría de las ocasiones nos encontraremos puñetazos o patadas, que podemos esquivar tal y como se ve en la imagen, colocándonos en el extremo izquierdo de la puerta a la hora de abrirla.

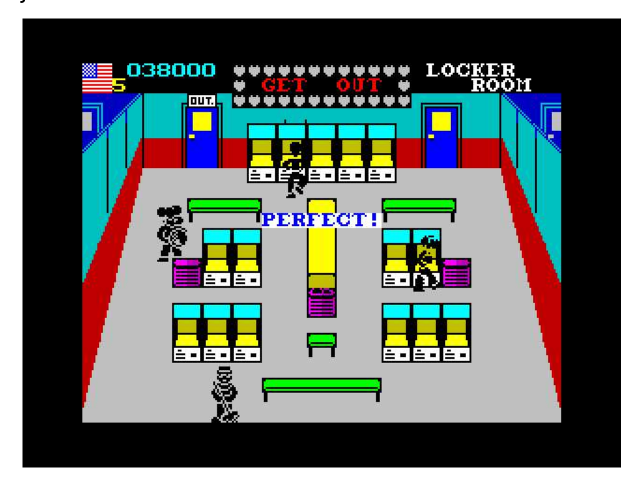
No todo son golpes, pues detrás de la puerta, si persistimos, aparecerá una chica que elevará la moral de nuestro intrépido playboy, reflejándose esto en los puntos con una bonificación extra.

En los pasillos tendremos que esquivar todo contacto con el conserje y el profesor. Este último, como en cualquier dependencia, acelerará su marcha cuanto más tiempo permanezcamos en la misma estancia. Así que no conviene demorarse mucho. A estos perseguidores tenemos que añadir el cubo del conserje, que de repente aparecerá desde cualquier extremo, a toda velocidad, y si nos pilla en medio nos restará una vida. En este primer pasillo sale siempre desde la derecha.

El pasillo cuenta con tres niveles en vertical. En ocasiones, el profesor y el conserje se situarán en el mismo nivel, lo que, salvo que utilicen el nivel central, nos deja cierta movilidad.

En ocasiones nuestros perseguidores se separan en dos niveles del pasillo. En estos casos, siempre que la distancia entre ambos nos lo permita, lo mejor es situarnos en el nivel extremo (superior o inferior) ocupado por uno de ellos, y hacer el cambio de nivel al pasar al otro perseguidor. No es un método 100% seguro, pero sí es cierto que funciona en la mayoría de las ocasiones.

Otro truco que leí en c.s.s. decía de cambiar rápidamente de posición entre los dos niveles ocupados. Sin embargo, en mi experiencia práctica el porcentaje de error es superior al anterior método, así que dejo esta posibilidad solo para cuando la separación de niveles entre profesor y conserje es dejando el nivel central libre.



El vestuario

La siguiente estancia en la que tendremos que recoger los corazones para nuestra amada es el vestuario. Aquí los corazones se encuentran dentro de las taquillas, y para cogerlos tenemos que situarnos enfrente de la taquilla correspondiente, mirando hacia ella, y gritarle una vez por cada corazón que tenga.

Igual que en la clase, se nos dará una puntuación extra si cogemos los corazones cuando estén parpadeando, con la diferencia de que en este caso cuenta el punto extra cuando es el último corazón de la taquilla el recogido cuando parpadea (los demás da igual si lo hacen).

En color rosáceo hay unos cestos donde se guardan balones de baloncesto. Estos balones podemos cogerlos con el disparo, y lanzárselos al perseguidor

que esté más cerca. Si lo coge se pondrá a botarlo por unos instantes, quedándose quieto en su sitio, con la ventaja que eso conlleva.

En esta estancia la bonificación extra se encuentra en las taquillas de la parte superior. Concretamente en la segunda taquilla empezando a contar desde la izquierda. Nos situamos frente a ella, la gritamos por tres veces y ya tenemos la bonificación.

Si nos esforzamos por hacer un nivel perfecto en esta fase podemos encontrar un fallo de programación, que en el caso más suave se resuelve sin demasiadas complicaciones.

Resulta que si conseguimos hacer el perfecto, en este nivel no se iluminará el cartel de la puerta de salida, quedándonos aparentemente atrapados en la habitación. En realidad resulta que por un fallo de programación, se contabiliza un corazón de menos en la última taquilla en la que los hemos recogido. Solo tenemos que volver a esa taquilla y hacer lo mismo que para recoger un corazón, que subirá al marcador, y nos iluminará el cartel de salida.



El segundo pasillo

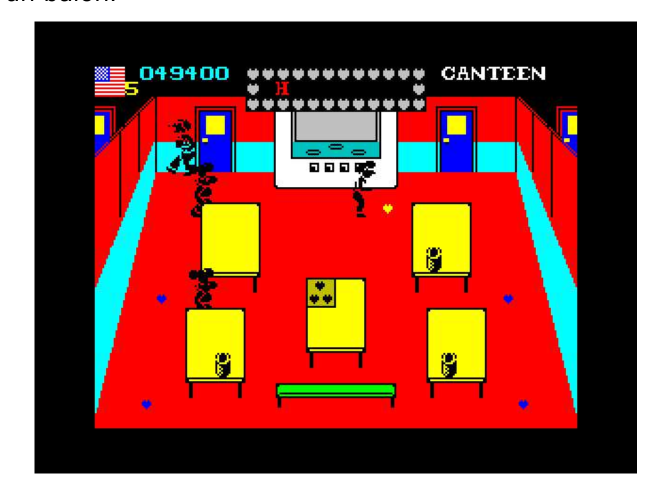
En el peor de los casos, todo se resolverá igual, pero de alguna manera extraña no funcionarán las máscaras de los sprites, vamos, que iremos marcando rastro por toda la pantalla, haciendo el juego casi impracticable. De todas formas, es raro que suceda esto último.

Poca novedad hay en este pasillo respecto al primero. En esta ocasión tenemos que bajar por las escaleras, diferenciándose la versión de Spectrum de su homónima en arcade en el hecho de que en nuestro Speccy el profesor se queda arriba, y el conserje esta siempre abajo, y en la recreativa nos persiguen también por la escalera.

En cualquier caso, los consejos para el primer pasillo son válidos para este pasillo, así que sin más pasamos a comentar la siguiente dependencia.

Aquí estarán persiguiéndonos dos cocineros y el profesor. Tenemos que recoger los corazones repartidos por el suelo, y los que están en la mesa. Estos últimos se recogen como los que se encontraban en las taquillas del vestuario.

En tres de las cinco mesas vemos una especie de botes, que en realidad por su contenido deben ser más bien ollas. En ellas hay tres pollos asados, que podemos recoger igual que hacíamos con los balones en el vestuario, y lanzárselos al perseguidor que este más cercano. Éste, si lo recoge, se pondrá a comerlo, quedándose igual de quieto que si estuviera botando un balón.



El comedor

Al igual que sucedía en el vestuario, podemos tener el problema de que, al recoger el último corazón de la pantalla, no se activa el indicador de la puerta de salida. La solución es la misma que antes, estando en esta ocasión el corazón oculto en la mesa del centro, donde estaban los demás corazones.

Como no podía ser menos, en el comedor hay también una bonificación oculta. Esta vez se esconde en la zona de dispensación de la comida, donde vemos los platos en la parte superior, en color cyan oscuro. Nos situamos frente al plato del centro, mirando hacia él, y gritamos tres veces. La puntuación extra subirá al instante a nuestro marcador.



El retorno al primer pasillo

Volvemos al primer pasillo, esta vez en las plantas inferiores. Y, como podemos apreciar en la imagen, no hay diferencias entre la forma de superar este pasillo y el segundo como lo hicimos hace un rato.

Cuando llegamos al gimnasio nos encontramos con una clase de baile compuesta únicamente por chicas. Aquí sólo nos va a perseguir el profesor, aunque no podemos despreciar el efecto que un simple roce con las chicas produce en Mikie: lo atonta lo suficiente para que pueda llegar el profesor y atraparnos.



El gimnasio

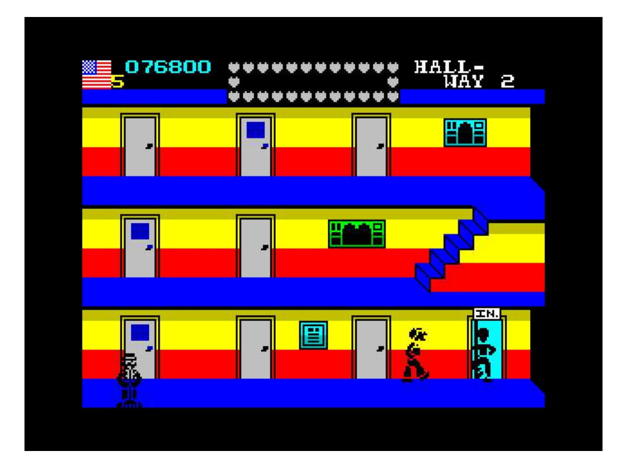
Recogemos los corazones con cuidado, teniendo en cuenta un detalle que nos resultará provechoso. Si caminamos pegados a las paredes esquivaremos siempre a las chicas, aun cuando se salgan del grupo para dar sus espectaculares pasos individuales.

La puntuación extra oculta en esta estancia está en el altavoz de la izquierda. Solo hay que situarse frente a él y, como hemos hecho hasta ahora, gritar tres veces.

Salimos al segundo pasillo otra vez, en esta ocasión en la planta inferior, sin disponer de escaleras, siendo esta vez similar al primer pasillo que superamos, en la tercera planta del primer pasillo.

Finalmente llegamos al jardín, donde nos espera nuestra enamorada. El profesor se ha quedado en el centro, así que aquí nos perseguirán tres conserjes a modo de jardineros.

Por tratarse de la última fase, aquí no tenemos puntuación extra, ni puerta. Solo a nuestra novia, que nos espera a que terminemos de recoger los corazones que quedan desperdigados por el jardín. Una vez los tengamos todos, nos acercamos por la izquierda a nuestra novia, entrando justo en el hueco que queda a su lado, y con esto finalizamos... la ronda.



El jardín

Sí, digo la ronda porque, una vez terminemos de celebrar haber llegado hasta la novia de Mikie, empezaremos de nuevo desde la clase, aumentándose la dificultad por dos partes: el número de corazones a recoger y la velocidad inicial del profesor.



Título	Jumping Jack
Género	Arcade/Platform
Año	1983
Máquina	ZX Spectrum 48K
Jugadores	1 Jugador
Compañía	Imagine Software Ltd
Autor	Albert Ball & Stuart C. Ball
Categoría en el Tour	Small Skill

Bueno, pues estamos ante un juego de habilidad sencillo de desarrollo, en el que podemos alcanzar altos niveles y buenas puntuaciones con algo de paciencia.

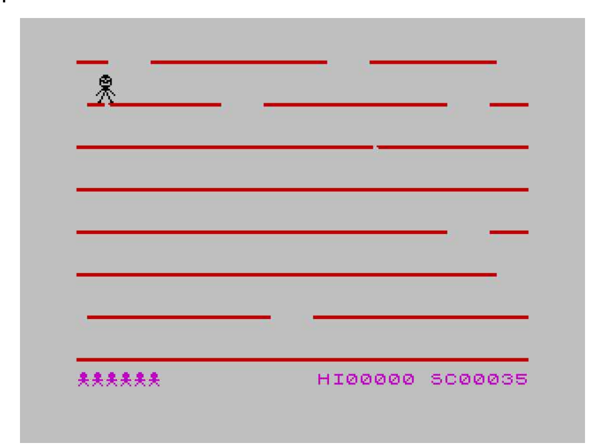
Como véis en la imagen, el juego se compone inicialmente de nuestro protagonista, Jack, y de ocho pisos. El objetivo del juego es alcanzar el último piso, saltando por los agujeros que se forman en el suelo, y que se desplazan entre ellos a izquierda o derecha.

La peculiaridad de estos agujeros es que, una vez llegan al extremo de la pantalla opuesto a por donde aparecieron, continuan desplazándose por el siguiente piso. Es decir, tenemos por ejemplo un agujero que se desplaza de derecha a izquierda. Cuando este agujero llega al extremo izquierdo del piso, aparece a continuación en el extremo derecho del piso superior, siguiendo con su recorrido. Si el piso es ya el más alto, entonces reaparece por el piso inferior, y así se repite indefinidamente. Si, por el contrario, el agujero se desplaza de izquierda a derecha, en lugar de continuar por el piso superior lo hace por el inferior.

La primera dificultad añadida al juego es el aumento en el número de agujeros. Cuando Jack se encuentra en la planta baja, por debajo del primer piso, solo hay un agujero que, por el azar del juego, seguirá una u otra dirección. Una vez saltemos a través de él alcanzando el primer piso, aparecerá un nuevo agujero en cualquier punto determinado por el azar,

siendo el sentido de circulación determinado por el mismo aspecto aleatorio.

Los agujeros solo sirven para saltar al piso superior. En ningún caso podemos saltar por encima de ellos sin llevarnos un golpe con el piso de arriba, que nos dejará muy aturdidos, pudiendo aparecer otro agujero en el piso en el que nos encontramos y caernos por el al piso inferior.



El primer nivel de Jumping Jack

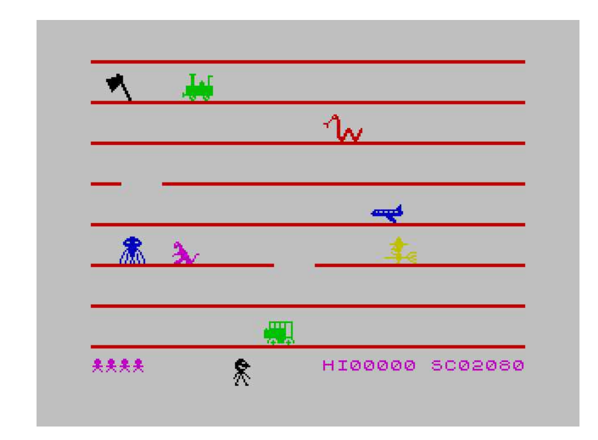
Hay que tener en cuenta un detalle, y es que Jack puede salir por los extremos de cada piso, apareciendo siempre en el extremo opuesto de por donde desapareció, siempre en el mismo piso. Es una buena forma de, en algunas ocasiones, esquivar agujeros y enemigos.

Jack no pierde vidas si se golpea con los pisos superiores, se cae por los agujeros o se golpea con los enemigos que aperecen en los niveles superiores. Tan solo resulta aturdido, sin podernos mover hasta pasado un rato en el que se recupera, si antes no hemos caido por un agujero que haya aparecido, o nos ha golpeado otro enemigo. Las vidas se pierden cuando caemos al nivel inferior.

El tiempo que Jack permanece aturdido depende de la forma en la que llega a este nivel. Lo que más le aturde es cuando se golpea con el techo del piso superior, seguido de los golpes con los enemigos, para ser el menos grave cuando se cae por uno de los agujeros.

Según vamos superando fases, van apareciendo enemigos en los diferentes pisos. Estos enemigos circulan todos a la misma velocidad hacia la izquierda, recorriendo los pisos de abajo a arriba como si fueran agujeros.

El número de enemigos aumenta a razón de un enemigo por cada nivel superado. La posición relativa entre ellos es siempre la misma, incluso entre niveles. Sin embargo, su posición relativa respecto a los agujeros varia por cada vez que consiguen llegar a la última planta, pues los agujeros reaparecen más tarde que ellos.



El número de enemigos aumenta

Considerando el comportamiento de los enemigos, el mejor truco para alcanzar el piso de arriba es utilizar un agujero que se dirija hacia la izquierda (vamos, de los que suben), y que disponga de espacio libre a su derecha equivalente al ancho de un piso. De esta forma, salvo por la aparición de los nuevos agujeros, el camino hacia la parte superior es fácil, únicamente perjudicado por la paciencia necesaria para esperar a que se dé la situación adecuada para subir, tiempo que suele aumentar a cada nivel.

El último nivel del juego no permite recurrir al truco indicado, pues no existe hueco entre los enemigos equivalente al ancho de un piso. En este caso todo lo que podemos esperar es que no se junten dos agujeros, y que *siempre* haya hueco suficiente a la derecha de los mismos, para que al menos al alcanzar el piso no seamos golpeados al instante por uno de los enemigos.

Y eso es todo lo que os puedo comentar del juego.



Título	Arcadia
Género	Arcade/Shoot'em Up
Año	1982
Máquina	ZX Spectrum 16K
Jugadores	1 Jugador
Compañía	Imagine Software Ltd
Autor	David H. Lawson
Categoría en el Tour	Small Shoot'em'up

Nos encontramos ante el último juego del Tour, y no por ello el menos interesante. Como es de esperar, no podía faltar un clásico matamarcianos, que en esta ocasión ha ocupado la categoría de pequeño shoot'em'up, no por ser de baja calidad, que no lo es, sino por su reducido tamaño. Y es que en la época

dorada del software de 8 bits, el tamaño si importaba.

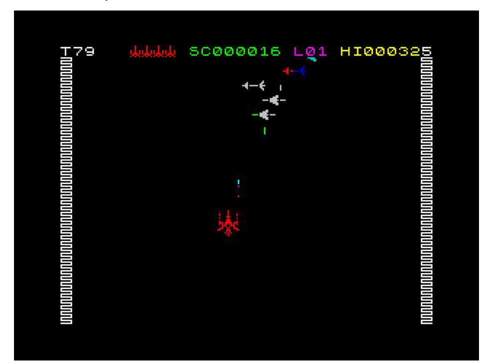
Bromas aparte, Arcadia representa a los clasicos arcades matamarcianos en los que con nuestra nave tenemos que ir acabando con todos los enemigos que aparecen por la pantalla.

Como era bastante habitual por la época, en lugar de seguir un scroll vertical u horizontal, el juego se divide en fases por pantallas, como en la recreativa Galaxian, pero con enemigos diferentes en cada pantalla.

El objetivo del juego no es acabar con todos los enemigos de cada pantalla, pues una vez hecho esto aparece una nueva oleada, y así hasta que se agote un temporizador que tenemos en la parte superior izquierda de la pantalla, que dura un minuto y medio aproximadamente. El objetivo es pues, aguantar todas las oleadas que podamos durante el tiempo indicado, finalizado el cual pasaremos a la siguiente fase.

Hay que tener cuidado, pues cuando el temporizador llega a cero todas las naves presentes en la pantalla se precipitan hacia abajo y, si nos pillan en medio, pues nos restan una vida, teniendo además que repetir la fase correspondiente. Vamos, que no hay nada como despejar un hueco en vertical para tener ciertas garantias de que cuando se precipiten no impactaran contra nuestra nave.

El juego no tiene final, siendo uno de estos juegos en los que una vez superas la última fase, vuelves a empezar desde el principio. Y lo que es mejor, en este caso, sin que apenas se perciba un aumento en la dificultad. Si a todo esto le añadimos que cada cuatro fases nos dan una vida extra, tenemos el juego perfecto para tirarnos horas y horas una vez le pillemos el truco... si las muñecas aguantan (y lo digo por experiencia).



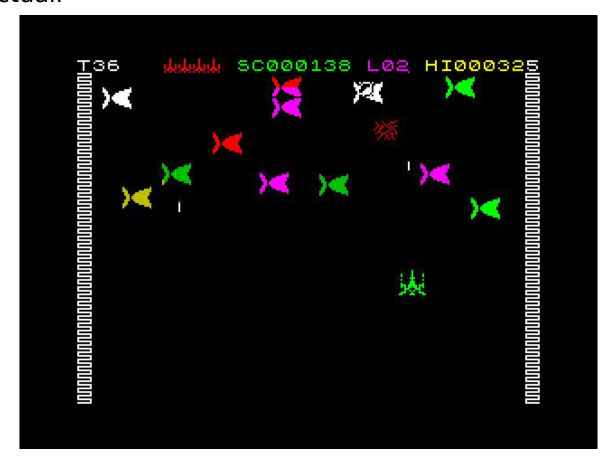
Los enemigos de la primera fase

El sistema de puntuación es de progresión aritmética. En la primera fase cada nave enemiga eliminada cuenta un punto, en la segunda, dos, en la tercera tres, y así sucesivamente fase tras fase. No es difícil darse cuenta el incremento de puntuación que significa tan solo una ronda más.

ARC-1. El comportamiento de los enemigos, o consejos para superar cada nivel

Lo primero que debemos tener en cuenta es que los enemigos *nos* disparan a la vez que recorren la pantalla según una trayectoria predefinida con la intención de chocar contra nosotros. Vamos, que son un poco kamikazes. Sabiendo eso y conociendo el patrón de trayectoria que van a seguir tenemos amplias posibilidades de superar al menos una vuelta a todo el juego.

En la primera fase los enemigos son un grupo de naves que aparece realizando un recorrido descendente diagonal hacia la izquierda desde una ubicación próxima a donde esta el indicador del nivel actual.



Los originales enemigos de la segunda fase

Hay muchas formas de superar esta fase, siendo mi preferida situarme un poco más a la izquierda de donde aparecen por pantalla las naves enemigas, lo más arriba que se nos permite subir, y quedarme disparando. Cuando aparecen las naves, se deja caer nuestra nave, y con un poco de suerte habremos eliminado todas las naves enemigas. Si queremos aumentar la probabilidad de exito, podemos dejarnos caer hacia la izquierda, siguiendo la trayectoria de las naves.

En la segunda fase las naves enemigas descienden en vertical, haciendo un ligero zigzag con el que se desplazan horizontalmente.

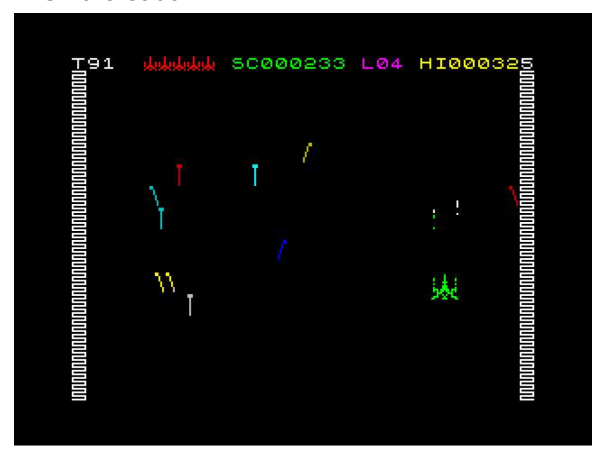


Los platillos volantes de la tercera fase

La manera que recomiendo para superar esta fase parte del mismo principio que la anterior, es decir, subiendo la nave lo más alto que podamos. Según aparezcan las naves enemigas, empezamos a disparar de forma contínua, a la par de dejarnos caer, con un ligero zigzag sin desplazarse en lo horizontal de manera absoluta. De esta forma creamos un hueco en la barrera enemiga consiguiendo así evitar que impacten contra nosotros.

Con el hueco creado podemos tratar de eliminar el resto de naves enemigas de la oleada, sin arriesgar demasiado. Solo hay que tener presente que cuando acabemos con la oleada, aparecerá otra nueva hasta que agotemos el tiempo.

En esta tercera fase aparece un grupo de platillos volantes cuyo movimiento se caracteriza por oscilar en torno a un punto que se desplaza horizontal y verticalmente de la misma manera oscilante, manteniendo siempre la formación dentro de una misma oleada.



Las agujas

Acabar con la primera oleada resulta bastante fácil pero, aunque no es una maniobra respetada por algunos jugadores, no lo recomiendo. Esto se debe a que la segunda oleada que aparece puede hacerlo con un desplazamiento brusco vertical, impactando alguna de las naves directamente contra la nuestra. Y claro, mejor aguantar con una nave de la primera oleada hasta que se agote el tiempo que arriesgar una vida... Salvo que ya dispongamos de unas cuantas.

Al superar esta fase obtenemos una vida extra.



Los pájaros

Las naves de esta fase tienen una forma alargada, como de aguja, y se precipitan hacia abajo a toda velocidad, siguiendo a la par líneas irregulares en zigzag en la horizontal.

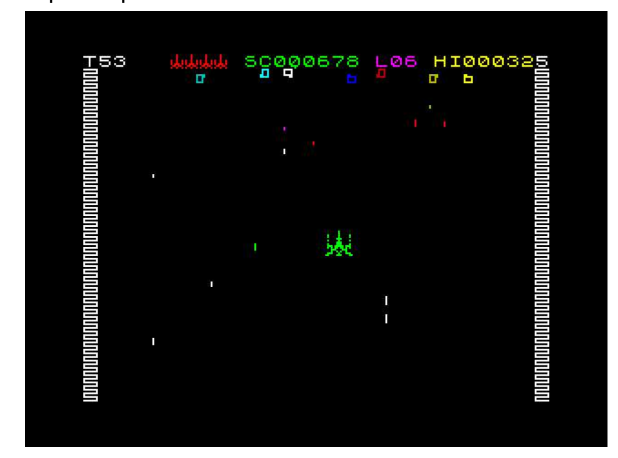
A mi juicio son de las más puñeteras, pues a la velocidad con la que se desplazan hay que añadir que su pequeño tamaño hace que nuestros disparos a veces pasen de largo.

La mejor manera de pasar esta fase es eliminar un grupo de 'agujas' que dejen un hueco vertical, independientemente de aguantar con la primera oleada o arriesgándonos con varias.

En esta fase las naves se asemejan a unos pájaros que, mientras descienden hasta nuestra posición, se dedican a dispararnos.

La eliminación de estos 'pájaros' no entraña ningún misterio, ya que como permanecen casi sobre la misma vertical, acertarles y eliminarles es disparar y mirar.

Ésta es una buena ocasión para aprovecharse y eliminar cuantas oleadas, las que nos permita el tiempo disponible.



Las hormiguitas

Las he llamado hormiguitas porque van todas juntas, en fila. Se desplazan haciendo un barrido de la pantalla de izquierda a derecha y de derecha a izquierda, bajando un poco en cada recorrido horizontal completado. Además no se cansan de dispararnos una y otra vez.

El modo de acabar con estas hormiguitas es situándose en lo más alto posible y, con el cuidado de esquivar sus disparos, comenzar a disparar sin soltar el botón del disparador por un solo instante.

Cuando son pocas hormigas las que quedan en pantalla, puede resultar algo más complicado acertarlas. En estos casos es buen truco esperar a que estén a tiro y, para asegurarnos el impacto, disparar en ráfaga mientras nos desplazamos en el mismo sentido en que lo hace la nave enemiga.



Las antenas

En cuanto a las oledas, dado que estas naves forman una barrera horizontal completa, conviene acabar tan

solo con una oleada, dejando una nave de la segunda oleada para tenerla controlada y esquivarla cuando el temporizador vaya a llegar a cero. De todos modos, hay que ser muy hábil para que el tiempo restante cuando dejemos solo una nave de la segunda oleada sea considerable.

No sé a vosotros, pero a mí las naves de esta fase me recuerdan a ciertas torres de antenas. Será deformación profesional.

En fin, las naves de esta fase se desplazan hacia abajo siguiendo un zigzag horizontal, al estilo de las de la segunda fase, pero con un zigzag más amplio y una mayor diferencia en la posición vertical relativa entre naves.

Como venimos haciendo hasta ahora, al comenzar la fase subimos la nave al máximo permitido y comenzamos a disparar, dejándonos caer a la parte inferior mientras que con un movimiento oscilante en la horizontal despejamos la vertical.

Ésta, como la de los pájaros, es una buena fase para puntuar eliminando tantas oleadas como nos plazca. Eso sí, siempre teniendo presente el tiempo restante indicado en el temporizador.

Al superar esta fase obtenemos una vida extra.



Los conos

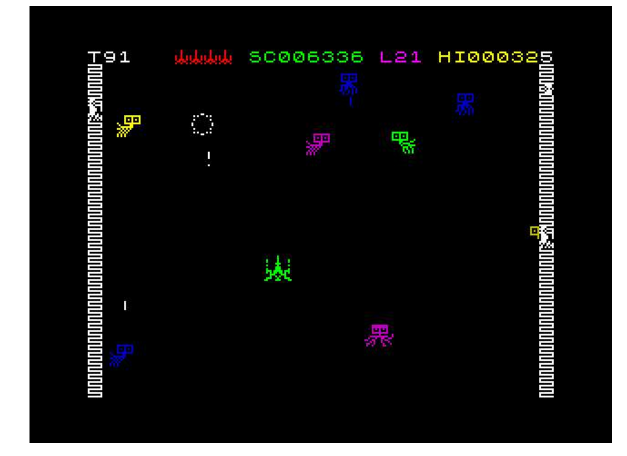
Por llamarlas de alguna forma, a las naves de esta fase las llamaremos los conos.

El movimiento que describen los conos en la pantalla es una gran variación respecto a los vistos hasta ahora. Se despliegan desde la parte superior describiendo trayectorias diagonales hacia izquierda y derecha, manteniéndose siempre en la misma trayectoria.

Superar esta fase ya requiere cambiar un poco de táctica. Pero sólo un poco pues, como hasta ahora, para mi gusto lo mejor sigue siendo elevar la nave y empezar a disparar. Tenemos que procurar situarnos donde cae la parte izquierda del marcador de puntos. Desde ahí, empezamos a disparar, manteniéndonos esta vez arriba, esquivando hacia izquierda y derecha las naves que van apareciendo.

Siguiendo esta estrategia podemos ir eliminando poco a poco las naves, despejando el camino y bajando ya a la parte inferior cuando veamos que va quedando hueco libre.

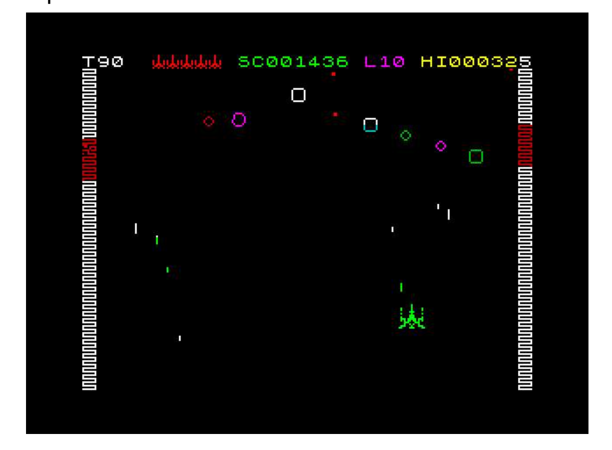
Es fácil caer en la tentación de eliminar la oleada completa y, bueno, mermar la segunda oleada y dejar el hueco libre es fácil una vez se le coge el truco a esta fase, así que una vez hayáis practicado, no lo dudéis, a por la segunda oleada.



Esos raros hexópodos

Llegamos a una fase caracterizada por unas naves realmente incordiantes. No sólo no siguen una pauta firme, si no que además tienen la mala costumbre de dividirse en grupos que suben y grupos que bajan.

En sí estas naves realizan un recorrido cuadrangular. Vamos, que o bien van hacia la izquierda, hacia arriba, hacia la derecha y luego hacia abajo, o bien hacia la izquierda, hacia abajo, hacia la derecha y hacia arriba, o los otros dos movimientos especulares que quedan.



Los destellos

Aquí poco truco puede haber más allá de mantener la nave siempre en alto, por aquellas naves que aparecen también desde abajo, y lo más centrada posible, por las que se cuelan desde los laterales.

Al superar esta fase obtenemos una vida extra.

Las naves de esta fase forman en fila, creando una barrera horizontal. Dentro de la fila, se mueven describiendo una trayectoria hazia la izquierda, luego hacia abajo, después a la derecha y finalmente hacia arriba. Y así hasta llegar a la parte inferior de la pantalla.

Como siempre, para ganar tiempo, el consejo es situarse en lo más alto de la pantalla que se nos permite, comenzando por uno de los extremos, a gusto del consumidor, y empezar a disparar, desplazándonos hacia el otro extremo según vayamos eliminando naves enemigas, y con cuidado de dejarlas pasar si están muy próximas a la parte inferior de la pantalla.

El número de oleadas ideal para esta fase es de dos, aunque se puede llegar a una tercera. Todo depende de nuestra habilidad para aprovechar el tiempo disponible.

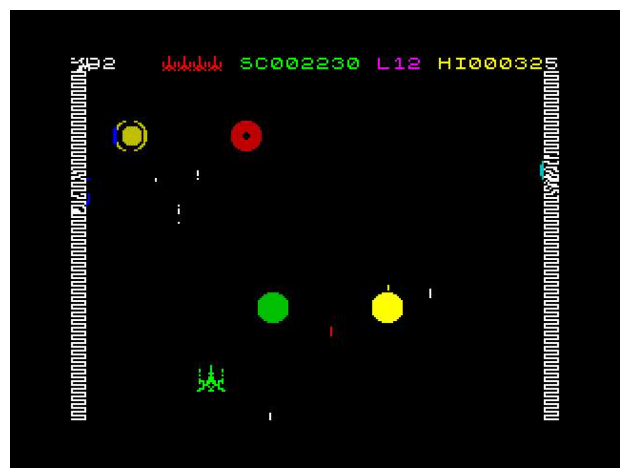


Los otros pájaros

En esta fase las naves parecen de nuevo unos pájaros, pero esta vez más grandes.

En esta ocasión las naves, cuando aparecen, realizan picados hacia la parte inferior de la pantalla, permaneciendo a una altura próxima al borde, no siempre la misma, esperando a que pasemos sobre la vertical de su posición. Cuando lo hacemos, desaparecen por la parte inferior para reaparecer desde arriba con un nuevo picado, quedándose esta vez esperando por la zona intermedia de la pantalla. De nuevo, al pasar por su vertical, desaparecen picándose hacia abajo, para reaparecer por arriba y quedarse casi por donde salen, de nuevo esperando a que pasemos por su vertical.

Las oleadas en esta fase son aleatorias, llegando al punto de que pueden aparecer pájaros en cualquier momento.



La fase final, las esferas

Podría parecer que la mejor forma de eliminar a los pájaros es dejando que llegen a la parte inferior, pasando por encima de ellos, dejando así que al reaparecer se queden en el centro de la pantalla y dispararlos entonces. Pero sucede que aparte de que pueden aparecer nuevos pájaros que nos ocupen la parte inferior, no dejándonos hueco posible entre la línea media y la inferior, cuando se lanzan en picado, si no tenemos suerte en el disparo, puede esquivarlo y acabar impactando su nave contra la nuestra.

En realidad no hay una estrategia definida para superar esta fase. Por mi parte, me suelo situar arriba, por la zona izquierda del marcador de puntos, disparando constantemente, ya que parece ser la zona predilecta para que aparezcan los pájaros. De esta forma elimino, con algo de suerte, todos los pájaros antes de que sean visibles.

Cuando se hacen visibles, me dejo caer tratando de destruirlos antes de que lleguen al nivel inferior. Y si consiguen llegar, entonces recurro al método que criticaba hace un momento.

Las esferas que hacen de naves enemigas en esta última fase hacen un recorrido curioso y a veces peligroso. Aparecen desde arriba, como es habitual, precipitandose hacia la parte inferior de la pantalla. A una altura aleatoria, cada esfera gira de repente, y se precipita horizontalmente a izquierda o derecha, para antes finalmente girar de nuevo y precipitarse otra vez hacia abajo hasta desaparecer, para luego reaparecer por arriba.

La estrategia a seguir en esta ocasión pasa, de nuevo, por subir lo más alto posible y disparar como un poseso. Hay que mantenerse en lo alto, esquivando las esferas según caen, salvo que alguna de ellas empiece a girar horizontalmente a nuestra altura, en cuyo caso habrá que hacer un quiebro bajando, pasando por debajo de ella y subiendo de nuevo. Si es posible al pasar por debajo de ella la disparamos, a ver si de paso la eliminamos.

Una vez superemos esta fase obtendremos una nueva vida extra, y pasaremos a la fase 13, que será una copia de la 1. Después seguira la fase 14, que será copia de la 2, y así sucesivamente.

ARC-2. Comentario final sobre Arcadia: '¿Tourable' o 'no tourable'?

Bien, he de reconocer que siendo esta mi primera participación en el Tour, puede que no esté del todo muy centrado en este comentario, si bien es un tema que se discutió, no sin cierto acaloramiento, en los foros de World of Spectrum y de comp.sys.sinclair.

Entre otras características que los organizadores del Tour tienen en cuenta a la hora de proponer los juegos candidatos para cada edición, una de ellas es evitar que el juego sea maratoneable. Y bien, ¿qué es eso de maratoneable?

Se dice de un juego que es maratoneable (venga, otra vez) cuando se dan las condiciones necesarias para que el juego no tenga fin y se convierta en una escalada de puntos para los jugadores más hábiles, pudiendo así descompensar las puntuaciones finales de la tabla en el Tour.

En el caso de Arcadia estas condiciones se daban en toda regla, pues las fases se repiten en ciclos de 12 niveles, sin apenas aumento palpable de la dificultad, con el añadido de que cada cuatro niveles superados se obtiene una vida extra.



SIEW

El Spectrum Hoy

Hay páginas que han estado con nosotros desde hace mucho tiempo, no solo porque ofrezcan contenidos en un principio interesantes sino porque incluso fomentan la participación de los aficionados al mundillo. Una de ellas es la conocida "El Spectrum Hoy", de Digiexp, hogar de las ultracargas, las competiciones de partidas grabadas, y las historias de Tony Brazil y Wally Week.

Todavía se puede recordar como hace bastante tiempo, un entonces novatillo en el mundo de las news, con alias Digiexp, decidió crear una pequeña web donde explicar cómo conseguía que los juegos tardaran casi nada en cargar en un Spectrum real. Más tarde vinieron las competiciones de partidas grabadas con el emulador RealSpectrum, con un nivel de participación moderado pero sin duda muy entretenidas (el autor de este texto no puede evitar emitir un alarido de rabia al pensar con qué maestría le *robaron* el primer puesto en la competición de Abu Simbel). Tiempo después Tony Brazil empezó a darnos sus opiniones desde la web y, desde ese momento, cada vez más y más secciones se iban añadiendo.

En esta ocasión vamos a analizar esta conocida página... ¿qué nos ofrece? ¿Cuáles son sus puntos fuertes y sus puntos débiles? Veámoslo en detalle.

DISEÑO

Es una lástima empezar por este apartado, porque sin duda el diseño es lo que más flaquea en la página de 'El Spectrum Hoy'. La web sigue el conocido esquema de un frame izquierdo más estrecho para poder acceder a las distintas secciones de la página, y un frame, mucho más ancho, a la derecha, para mostrarnos el contenido en sí mismo. Por lo tanto, al pulsar sobre cada una de las secciones a la izquierda, en la parte derecha veremos qué nos ofrece. En la sección de partidas grabadas, además, se genera un frame superior para acceder a cada una de las distintas subsecciones... pero aquí es donde encontramos el primer fallo grave de diseño (al menos en el navegador Konqueror, que es el que utiliza el autor de este texto): al pulsar sobre una de las subsecciones superiores, veremos como en el frame central, en el que se suele mostrar el contenido, se vuelve a crear un frame superior. Si volvemos a pulsar sobre una subsección del nuevo frame superior, se volverá a crear un nuevo frame superior en el frame central, y así hasta que queramos. Sin duda es algo que se debería subsanar para que las personas que visiten la página no se lleven una mala impresión.



Pagina Principal de El Spectrum Hoy

Los colores de la web tampoco son demasiado atractivos. No es incómodo ver un fondo negro con letras blancas, pero cuando la fuente empieza a cambiar de color sin ton ni son, el resultado es bastante extraño. Esto se hace patente en secciones como la de ultracargas, donde las combinaciones de colores, en opinión del autor, no son precisamente

atractivas visualmente. Los parrafos también quedan muy separados en varias ocasiones, dando sensación de vacío; y se combinan párrafos centrados con párrafos no centrados, dando sensación de desorden... en definitiva, toda la cuestión de aspecto visual en lo referente a colores y párrafos debería ser retocada (en opinión del autor).

Por último, decir que en la sección de Grandes Desconocidos, es un poco incómodo que aparezcan todos en la misma página, porque tienes que esperar a que carguen todas las imágenes para poder disfrutarlas. Aprovechando pues que ya existen enlaces al principio de esa sección para cada uno de los juegos mostrados, sería interesante que cada uno de dichos enlaces, en lugar de trasladarnos a otra localización de la misma página, nos guiaran a páginas distintas (como se hace por ejemplo en la subsección de Partidas Acabadas dentro de la sección de Partidas Grabadas).



Fallo grave de diseño en los frames en la sección de Partidas Grabadas

Por lo tanto, nos encontramos ante una web con un diseño un tanto estrambótico, pero... ¿y los contenidos? ¿Son tan irregulares?

CONTENIDO

En la página principal siempre ha habido sitio para hablar de las ultracargas, dejándonos claro que El Spectrum Hoy es la residencia oficial de este novedoso método de carga. Además, se suelen poner allí las novedades más impactantes de la web (en el momento de escribir esto, sin duda alguna, la entrevista a Bo Jangeborg, creador de Artist y Fairlight). Sólo por estas dos cosas, ya vale la pena echarle un vistazo a la web. En la sección de Noticias ya podremos consultar de forma más detallada las novedades, así como la sección donde podremos encontrarlas.

La sección de Manuales contiene información bastante interesante; documentos que explican cómo realizar las ultracargas (luego hablaremos de ellas), cómo grabar partidas en algunos emuladores (imprescindible para los novatos que quieran participar en competiciones), algo sobre el Spectrum en la Dreamcast, o como conectar una fuente de sonido externa al Spectrum. No son cortos precisamente, y son bastante curiosos, sin duda interesantes, aunque se centra demasiado en las herramientas para Windows.

La sección del Speccy Tour tiene su razón de ser en que durante la última edición de esta conocida competición de partidas grabadas, la página de El Spectrum Hoy fue 'sede oficial'. Muestra un histórico de las partidas del 2002 y del 2003... ¿seguirá en años posteriores? ¿Recuperará las partidas de años anteriores?

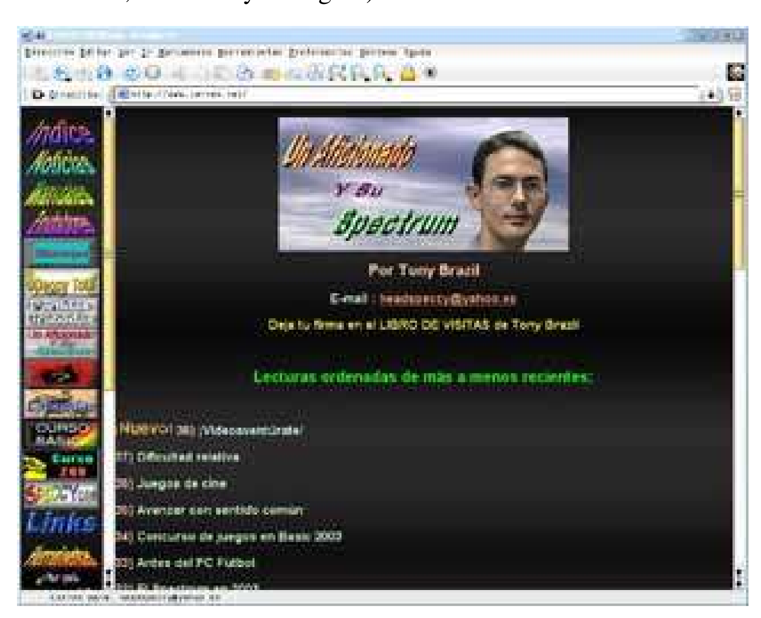
La sección de Ultracargas fue la que originó esta página. El método de la ultracarga fue desarrollado por Digiexp para permitir que los juegos tardaran mucho menos tiempo en cargar, ya fuera desde cinta, o desde una fuente digital como el CD. Podemos acceder a una inmensa cantidad de ultracargas, de tal forma que solo tendremos que esperar uno o dos minutos, e incluso a veces menos, para poder jugar a los juegos. En la sección de manuales todo esto se explica en detalle, y es altamente recomendable.

La siguiente sección es una de las más conocidas en esta web, la de Partidas Grabadas. Puedes grabar tus partidas con un emulador y enviarlas para cada una de los apartados de esta sección. Por ejemplo, en Partidas Acabadas podemos ver cómo la gente termina un juego, desde que comienza la partida hasta que se nos muestra el mensaje final de felicitación. En Partidas de Competición el objetivo no es mostrar cómo se supera un juego, sino que en enviar partidas con la máxima puntuación posible para ver si otros te superan. En Partidas Alternativas tenemos partidas curiosas o extrañas que no tienen cabida en los dos apartados anteriores. Por útlimo, tenemos el apartado de competición. Cada cierto tiempo Digiexp proponía un juego para competir sobre él y la gente iba enviando sus partidas hasta un tiempo límite, hasta conseguir un ganador. Hablamos en pasado porque la última competición, la de Nebulus, fue hace bastante tiempo, y la verdad es que el resto de apartados de la sección de Partidas Grabadas parecen bastante muertos. Es una verdadera lástima que la sección más divertida de la web no tenga una continuidad aparente. Es una sección que, como decimos, parece un poco muerta.

La siguiente sección también es de las conocidas de la página, la de Un Aficionado y su Spectrum, donde Tony Brazil, habitual de las news de Spectrum en español, de forma incansable y regular nos va contando sus opiniones, recuerdos, hasta llegar en la actualidad a un total de 38 textos. Dichos textos nos recuerdan directamente a sus posts en las news de Spectrum, son muy parecidos. Una sección,

sin duda, para fans. Se agradecerían a veces comentarios más profundos y concretos. Por ejemplo, en el texto sobre Uchi Mata, Tony habla de juegos malos. Sobre el L.A. Swat nos dice: "Juas, este juego es que es malo y tonto a la vez. Carece de todo tipo de interés"; ¿por qué, Tony? No todo el mundo tiene por qué conocer el juego, o no todo el mundo tiene por qué tener la misma opinión, y esas cosas siempre es bueno razonarlas.

La siguiente sección se llama Grandes Desconocidos y, según se comenta en la web, está destinada a mostrarnos juegos que en su momento pasaron desapercibidos, pero que son lo suficientemente buenos como para que sean más extendidos. Sin embargo, aproximadamente la mitad de los juegos mostrados son posteriores al año 90, e incluso posteriores al año 93, por lo que no es de extrañar que en su momento pasaran desapercibidos, porque en esa época el Spectrum estaba muerto comercialmente y nuestra fiebre de recuperación del mito Spectrumnero todavía no había aparecido (e Internet no era lo suficientemente popular como para que pudiéramos conocer la existencia de estos juegos desconocidos). Se salva un poco por el hecho de que el resto ya sí que se encuadran dentro de lo que sería la época comercial de nuestra máquina, mereciéndose por lo tanto el calificativo de 'desconocidos en su época' (aunque algunos de ellos no pasarán tan desapercibidos, como el Scuba Dive, el Moi Reversi, o el Barmy's Burguer).



¿Qué nos contará hoy Tony Brazil?

La sección de Leyendas es parecida a la de Un Aficionado y Su Spectrum, solo que en este caso cualquier persona puede colaborar con sus textos y no solo Tony Brazil. Aunque, sin embargo, la inmensa mayoría de los textos son escritos por Wally Week, uno de los colaboradores de la web. Eso sí, cualquiera puede participar. La última entrada es la más importante, la entrevista a Bo Jangeborg que se comentaba anteriormente, de recomendada lectura.

Las dos secciones siguientes están un poco muertas. Son las del curso de Basic y la del curso de Z80. De hecho, en el curso de Z80 solo se incluyó un capítulo hace muchísimo tiempo y nunca más ha vuelto a ser actualizada. Son unas secciones que fueron anunciadas a bombo y platillo, pero que nos han decepcionado un poco, porque no han sido capaces de ofrecernos contenido de forma regular, o incluso en una de ellas, contenido en absoluto desde que fue creada.

Los siguientes enlaces del frame izquierdo nos envían a la página de Speccy.org, a la sección de enlaces de El Spectrum Hoy, y a una página que nos cuenta por qué Digiexp decidió empezar con este trabajo.

CONCLUSIONES

A la hora de puntuar esta página, se ha decidido separar en dos apartados (al contrario que en comentarios anteriores): diseño y contenido. Esto es así porque el diseño es bastante mejorable y muy poco atractivo y podría incidir muy negativamente en una puntuación global. Con respecto al contenido, se salva un poco, pero algunas secciones muertas hacen también que la nota sea baja.

Lo que más llama la atención (en un sentido positivo) de esta página es la sección de ultracargas y la de partidas grabadas. Lo que menos agrada desde un principio, y que puede hacer que mucha gente se eche para atrás a la hora de visitar la página, es el aspecto visual, que debería ser completamente retocado.

Sería conveniente una revisión de los contenidos para eliminar alguna sección, como la del curso de Z80, que no ofrece apenas contenidos y no se ha actualizado nunca, y reorganizar otras, como la de Grandes Desconocidos, para distinguir grandes desconocidos durante la época del Spectrum, y nuevas creaciones después de la época comercial de nuestra máquina. Los textos de Tony podría ser un poquito más rigurosos, sin necesidad de abandonar la subjetividad, pues a fin de cuentas es una sección de opinión.

Una página muy conocida que, tras un pequeño empujoncito, podría convertirse quizás en una de las mejores.

PUNTUACION: 3,5 (DISEÑO) - 6 (CONTENIDO)

LINKS

- El Spectrum Hoy : http://www.speccy.org/spectrumhoy/ o bien http://www.jaivan.net



SIEW

CREANDO UNA AVENTURA CONVERSACIONAL CON Z88DK (I)

Ha llegado el momento de hacer algo útil con la librería y, para ello, damos nuestros primeros pasos creando una sencilla aventura conversacional. Está claro que siempre existirán alternativas mejores para la creación de este tipo de aventuras en nuestro Spectrum (el PAWS, sin ir más lejos, u otros parsers). Sin embargo, una aventura conversacional puede ser un ejemplo didáctico sencillo y a la vez atractivo... ¡estaremos programando un juego desde un primer momento! Más adelante ya hablaremos de sprites, movimiento, colisiones, etc.

La historia

Como todo buen creador de aventuras conversacionales sabe, el primer paso para crear algo que interese al jugador es tener una buena historia de transfondo; sin ella, la aventura pierde interés. Debemos huir del típico argumento de salvar una princesa o conseguir un tesoro, y pensar en algo original. También debemos pensar en el tipo de público al que se dirige la aventura, entre otras cosas.

Como lo que tratamos de hacer nosotros es una aventura de ejemplo, que nos sirva para practicar con el z88dk, está claro que buscaremos sencillez; sencillez en la historia a contar, en el número de habitaciones a visitar y objetos con los que interactuar. Así pues, nuestra (poco original) historia es la siguiente:

Guybrush Threepwood quiere ser pirata. Para ello se ha dirigido a la isla de Melêe, donde le han dicho que un maestro le puede enseñar a serlo. La historia comienza en la taberna de la Isla de Melêe, donde Guybrush deberá buscar al maestro e impresionarlo con algún truco para que acepte enseñarle todo lo que sabe.

Con este sencillo guión inicial pretendemos construir una aventura con no más de seis habitaciones, tres objetos con los que interactuar, y un personaje pseudo-inteligente o PSI. Para ello haremos uso de la librería z88dk. Iremos viendo que, salvo algunas peculiaridades, es como si estuvieramos programando una aventura conversacional en modo texto en C para un PC actual, ya que nos limitaremos a hacer uso de las librerías stdio.h y string.h incluidas con z88dk, muy parecidas a sus contrapartidas en el mundo de los compiladores estándar actuales. Así pues, comencemos.

¿Qué necesitamos?

Por supuesto, el primer paso es tener z88dk instalado tal como se comentó en artículos anteriores. Lo siguiente es crear un entorno de trabajo donde tengamos el código fuente de nuestro juego conforme lo desarrollemos. Así pues, creamos un nuevo directorio allá donde queramos, para almacenarlo. No es necesario que el directorio se encuentre dentro de la carpeta donde hubiésemos instalado el z88dk, pues si le damos el valor correcto a las variables de entorno (por ejemplo, ejecutando el setpath.h que se propuso en la primera entrega de z88dk) podremos trabajar en cualquier lugar de nuestro disco duro.

Dentro de ese directorio de trabajo tendremos que crear los siguientes archivos:

- aventura.c: será el fichero que contendrá el código principal de nuestro juego, y por lo tanto, el método main.
- datos.h: un archivo de cabecera que utilizaremos para definir las estructuras de datos que se utilizarán para la creación de la aventura.
- y, opcionalmente, un archivo Makefile, si utilizamos la herramienta GNU Make para la construcción automática de nuestro programa.

En el caso de que no utilicemos Make, cada vez que queramos compilar el programa y probarlo con un emulador, deberemos teclear lo siguiente:

```
zcc +zxansi -vn -o aventura.bin aventura.c -lndos
bin2tap aventura.bin aventura.tap
```

El parámetro +zxansi indica que se trata de un programa de solo texto para un Spectrum. El parámetro -vn desactiva el

modo verbose, lo cual quiere decir que se nos mostrará la mínima información posible por pantalla durante el proceso de compilación. Y -lndos le dice al compilador zcc que enlace con los archivos de librería adecuados cuando hacemos uso de stdio.h y string.h (no tiene nada que ver con el sistema operativo MS-DOS).

Sin embargo, si disponemos de la utilidad Make, podemos crear un archivo Makefile como el siguiente:

```
all: aventura.tap

aventura.tap: aventura.bin
    bin2tap aventura.bin aventura.tap

aventura.bin: aventura.c
    zcc +zxansi -vn -o aventura.bin aventura.c -lndos
```

y ejecutar 'make' en el mismo directorio donde se encuentre este archivo Makefile cada vez que cambiemos el archivo aventura.c y queramos compilar.

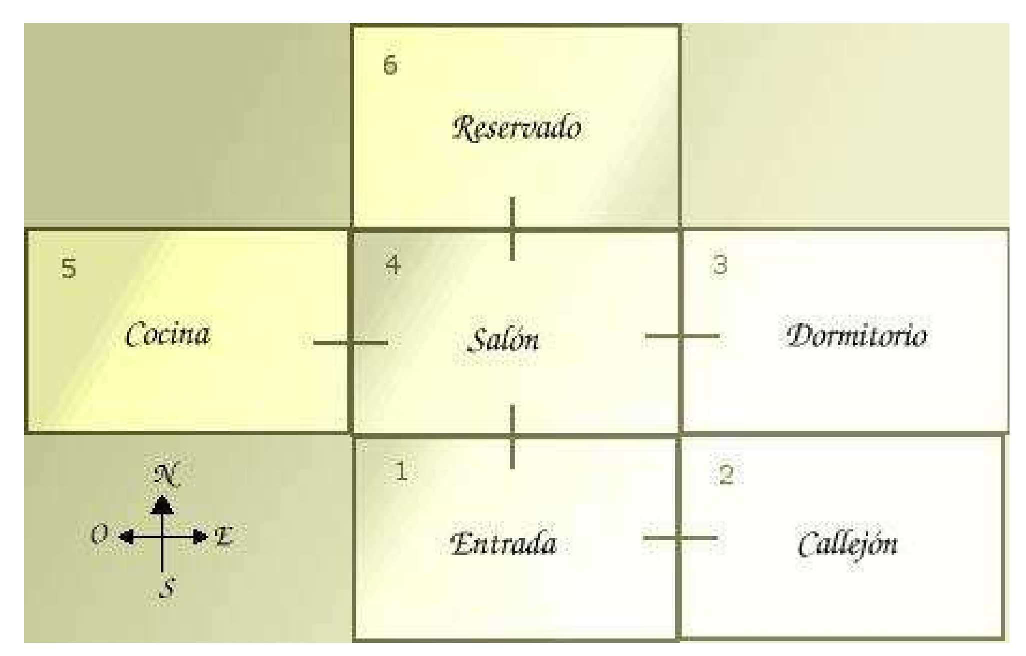
Creando las habitaciones

Para tener claro como va a ser el entorno de juego, es necesario dibujar un mapa. En dicho mapa deben estar claras cúales van a ser las localizaciones o habitaciones por las que va a poder desplazarse el jugador, y cómo van a estar conectadas entre sí. En el caso de nuestro ejemplo concreto, vamos a crear un mapa de la taberna de la isla de Melêe, en donde Guybrush, nuestro personaje, buscará a un maestro de piratería. Un mapa perfectamente válido podría ser el de la figura:

Como vemos, cada habitación está numerada. Esto es muy importante, luego veremos por qué. Las líneas que unen dos habitaciones adyacentes indican que dichas habitaciones están comunicadas; esto quiere decir que desde una habitación podremos acceder a la otra y viceversa. Además de dibujar el mapa, debemos escribir una descripción para cada habitación. Para nuestra aventura concreta, podríamos usar las siguientes descripciones:

- Habitación 1: "Te encuentras en la entrada de la taberna de Melêe Island. Desde aquí puedes oler los vómitos de los borrachos que vas a encontrar en su interior."
- Habitación 2: "Las sombras del callejón en el que te encuentras te sobrecogen. Estás rodeado de montones de basura."
- Habitación 3: "Estás en uno de los dormitorios de la taberna. Varias literas están alineadas, permitiendo a muchos piratas dormir juntos en una misma habitación."
- Habitación 4: "El salón principal de la taberna. Decenas de piratas se encuentran aquí bebiendo, cantando, peleándose, vomitando, y metiendo mano a las doncellas del lugar. Si te diriges al norte, este u oeste podrás visitar varias de las habitaciones del local."
- Habitación 5: "Una gran cantidad de pucheros sucios están amontonados en la cocina. Platos a medio comer, jarras medio llenas, y cubiertos sucios les hacen compañía."
- Habitación 6: "Tras apartar una cortina accedes a un reservado. Es una zona exclusiva de la taberna a donde muy pocos piratas pueden acceder."

Una vez se ha realizado el diseño, es hora de realizar una implementación de todo esto. Debemos crear una estructura de datos que nos permita almacenar la información sobre las habitaciones. Podemos incluir el siguiente código dentro de datos.h:



```
typedef struct
{
    int id;
    char descripcion[255];
    int direcciones[4];
            // 0 - Norte
            // 1 - Este
            // 2 - Sur
            // 3 - Oeste
} THabitacion;
```

La estructura THabitacion es la que nos va a permitir disponer de la información de las habitaciones de una forma ordenada. Cada uno de los campos va a ser utilizado para lo siguiente:

- id: identificador de la habitación. Cada habitación debe estar numerada desde el 1 en adelante, tal como se hizo con el mapa a la hora de diseñar la aventura. Es en este campo donde introduciremos el número de la habitación indicado en el mapa; en nuestro ejemplo particular, el id de la entrada a la taberna sería el 1, el id del callejón valdría 2, etc. En nuestra implementación es inútil, y además ocupa memoria (algo de lo que nuestro Spectrum no anda demasiado sobrado), pero se incluye a efectos ilustrativos.
- descripcion: una cadena de texto que contendrá la descripción de la habitación que será mostrada al jugador durante la aventura. Podemos usar los textos anteriormente indicados, teniendo en cuenta que no podremos utilizar tildes ni eñes (ni tampoco el acento circunflejo en el nombre de la isla de Melêe).
- direcciones: un array de cuatro elementos, representando cada uno de esos elementos una dirección concreta. En la posición 0 almacenaremos el identificador de la habitación hacia la que avanzaremos si nos desplazamos hacia el norte, en la posición 1 el identificador de la habitación hacia la que avanzaremos si nos desplanzamos hacia el este, en la posición 2 el identificador de la habitación hacia la que avanzaremos si nos desplazamos hacia el sur, y en la posición 3 el identificador de la habitación hacia la que avanzaremos si nos desplazamos hacia el oeste. Si no es posible desde la habitación desplazarse en algnuna de las direcciones especificadas, deberíamos almacenar el valor 0, indicando que no es posible que el jugador se mueva en esa dirección. Por supuesto, si quisieramos hacer una aventura más compleja con más direcciones de movimiento posibles, podríamos aumentar el tamaño de este array.

Introduciendo las habitaciones en nuestro juego

Ahora que ya tenemos nuestra estructura de datos que almacenará la información de las habitaciones, es hora de crearlas; ¡vamos a comenzar a programar!

Antes de nada, deberemos introducir las sentencias #include adecuadas en aventura.c; concretamente, al comienzo de nuestro archivo deberemos teclear lo siguiente:

```
#include "datos.h"

#include <stdio.h>
#include <string.h>
```

El primer include (datos.h) nos permite utilizar las estructuras de datos creadas en nuestro archivo datos.h. Se usan comillas porque es un archivo que se encuentra en el mismo directorio que aventura.c. Los siguientes includes nos proporcionan los métodos de entrada y salida y manipulación de cadenas de z88dk, que son los que vamos a utilizar en nuestro programa. Se introducen entre símbolos de mayor y menor para hacer ver al compilador que se deben buscar estos archivos de cabecera en la ubicación especificada por las variables de entorno y no en el mismo directorio que aventura.c.

Creamos nuestra primera función main, dentro de aventura.c, de la siguiente manera:

```
void main(void)
{
    int habitacion = 0;

    THabitacion habitaciones[6];
    inicializarHabitaciones(habitaciones);

    escribirDescripcion(habitaciones,habitacion);
}
```

Hemos declarado una variable llamada habitaciones que no es mas que un array de estructuras THabitación. Dicho array tiene seis posiciones, porque seis son las habitaciones de las que va a disponer nuestra aventura. La variable habitación hace referencia a la habitación en la que se encuentra actualmente el jugador. Así pues, si habitación vale 0, significará que nos encontraremos en la habitación con índice 0 del array de habitaciones (recordemos que en C, para un array de tamaño N, la primera posición tiene índice 0 y la última N-1). Así pues, al crear esta variable asignándole el valor 0, significa que el jugador inicialmente comenzará en dicha habitación (la entrada a la taberna de los piratas). Durante el transcurso de la partida, el valor de esta variable cambiará para indicar en que habitación se encuentra el

jugador. Es importante destacar que todas las variables locales a una función y que vayan a ser utilizadas en el interior de la misma deben ser declaradas en su inicio. No se pueden declarar variables intercaladas con el código de las funciones (como sucede con el C++).

Evidentemente, si intentamos compilar este código, obtendremos un error, porque hay dos funciones que no hemos implementado: inicializarHabitaciones y escribirDescripcion. La primera la vamos a utilizar para rellenar el array habitaciones con los datos correspondientes a las habitaciones de nuestra aventura. La segunda para mostrar el texto de la habitación en la que se encuentra el personaje por pantalla. La implementación de la primera función podría ser la siguiente (debe teclearse este código antes que la función main):

```
void inicializarHabitaciones(THabitacion habitaciones[])
{
    habitaciones[0].id = 1;
    strcpy(habitaciones[0].descripcion,"Te encuentras en la entrada
    de la taberna de Melee Island. Desde aqui puedes oler los vomitos
    de los borrachos que vas a encontrar en su interior.");
    habitaciones[0].direcciones[0] = 4;
    habitaciones[0].direcciones[1] = 2;
    habitaciones[0].direcciones[2] = 0;
    habitaciones[0].direcciones[3] = 0;

    habitaciones[1].id = 2;
    strcpy(habitaciones[1].descripcion,"Las sombras del callejon en el
    que te encuentras te sobrecogen. Estas rodeado de montones de
    basura.");
    habitaciones[1].direcciones[0] = 0;
    habitaciones[1].direcciones[1] = 0;
    habitaciones[1].direcciones[2] = 0;
    habitaciones[1].direcciones[3] = 1;

    habitaciones[2].id = 3;
    strcpy(habitaciones[2].descripcion,"Estas en uno de los dormitorios
    de la taberna. Varias literas estan alineadas, permitiendo a muchos
    piratas dormir juntos en una misma habitacion.");
    habitaciones[2].direcciones[0] = 0;
    habitaciones[2].direcciones[1] = 0;
    habitaciones[2].direcciones[2] = 0;
    habitaciones[2].direcciones[3] = 4;

    habitaciones[3].id = 4;
    strcpy(habitaciones[3].descripcion,"El salon principal de la taberna.
    Decenas de piratas se encuentran aquí bebiendo, cantando, peleandose,
    vomitando, y metiendo mano a las doncellas del lugar. Si te diriges
    al norte, este u oeste podras visitar varias de las habitaciones
    del local.");
    habitaciones[3].direcciones[0] = 6;
    habitaciones[3].direcciones[1] = 3;
    habitaciones[3].direcciones[2] = 1;
    habitaciones[3].direcciones[3] = 5;

    habitaciones[4].id = 5;
    strcpy(habitaciones[4].descripcion,"Una gran cantidad de pucheros
    sucios estan amontonados en la cocina. Platos a medio comer, jarras
    medio llenas, y cubiertos sucios les hacen compañia.");
    habitaciones[4].direcciones[0] = 0;
    habitaciones[4].direcciones[1] = 4;
    habitaciones[4].direcciones[2] = 0;
    habitaciones[4].direcciones[3] = 0;

    habitaciones[5].id = 6;
    strcpy(habitaciones[5].descripcion,"Tras apartar una cortina accedes
    a un reservado. Es una zona exclusiva de la taberna a donde muy
```

```
    pocos piratas pueden acceder.");
    habitaciones[5].direcciones[0] = 0;
    habitaciones[5].direcciones[1] = 0;
    habitaciones[5].direcciones[2] = 4;
    habitaciones[5].direcciones[3] = 0;
}
```

Esta función recibe como parámetro el array de habitaciones, que tiene seis posiciones, y para cada una de dichas posiciones inicializa la habitación correspondiente. Observemos por ejemplo como se inicializa la primera habitación, almacenada en la posición 0 del array. Con habitaciones[0].id = 1 indicamos que el identificador de esa primera habitación es 1. Esto realmente no va a ser utilizado en el código, pero para este ejemplo nos puede permitir ver las cosas más claras. Lo siguiente es, mediante el uso del método strcpy proporcionado por la librería de cadenas de z88dk, asociar la habitación con una cadena de texto que contendrá su descripción. Este método strcpy, al igual que en el C del PC, permite copiar en la cadena pasada como primer parámetro la cadena pasada como segundo parámetro, siempre que en la primera cadena haya espacio suficiente para ello. Las siguiente cuatro líneas almacenan a que habitación llegaremos si nos desplazamos en cada una de las cuatro direcciones posibles. Por ejemplo, si nos desplazamos en la dirección 0 (norte) llegaremos a la habitación 4. Si nos desplazamos en la dirección 1 (Este), llegaremos hasta la habitación 2. Para las direcciones 2 y 3 (sur y oeste) almacenamos el valor 0 indicando que no podemos desplazarnos en dicha dirección. El resto de habitaciones se inicializa de igual forma.

La siguiente función que debemos implementar para que nuestro código compile es escribirDescripcion, que como decíamos, presenta por pantalla la descripción de la habitación actual (aquella cuyo índice se almacena en la variable habitacion). Necesita dos parámetros: el array conteniendo la información de todas las habitaciones, y el índice de la habitación actual. El código de esta función podría ser el siguiente (como en el caso anterior, debe ser introducido en el código antes que el método main):

```
void escribirDescripcion(THabitacion habitaciones, int habitacion)
{
    printf(habitaciones[habitacion].descripcion);
    printf("\n\n");
    printf("Salidas:");
    if (habitaciones[habitacion].direcciones[0] != 0)
        printf(" %c[4mNorte%c[24m",27,27);
    if (habitaciones[habitacion].direcciones[1] != 0)
        printf(" %c[4mEste%c[24m",27,27);
    if (habitaciones[habitacion].direcciones[2] != 0)
        printf(" %c[4mSur%c[24m",27,27);
    if (habitaciones[habitacion].direcciones[3] != 0)
        printf(" %c[4mOeste%c[24m",27,27);
    printf("\n\n");
}
```

La primera línea no hace más que mostrar por pantalla la cadena de texto de la habitación actual que está almacenada en el array habitaciones. Se hace uso de printf, proporcionado por la librería de entrada y salida estándar de z88dk. La función printf se utiliza igual que en la librería estándar de C para PC, incluyendo el escape de carácteres como el \n (que introduce un retorno de carro), o la escritura de variables mediante %c, %d, etc.

La siguiente línea introduce dos retornos de carro para hacer el texto más claro, y la siguiente muestra la cadena "Salidas:", pues además de mostrar por pantalla la descripción de la habitación, vamos a indicar en que direcciones se podrá mover el jugador. Para cada posición del array direcciones para la habitación correspondiente, comprobamos si el valor es cero o no. Recordamos que si el valor era cero significaba que el jugador no podía moverse en esa dirección, y en caso contrario, el valor almacenado era el identificador de la habitación hacia la que nos desplazamos moviéndonos en esa dirección. En el caso de que para una dirección concreta el valor del array sea distinto cero, mostraremos la dirección correspondiente.

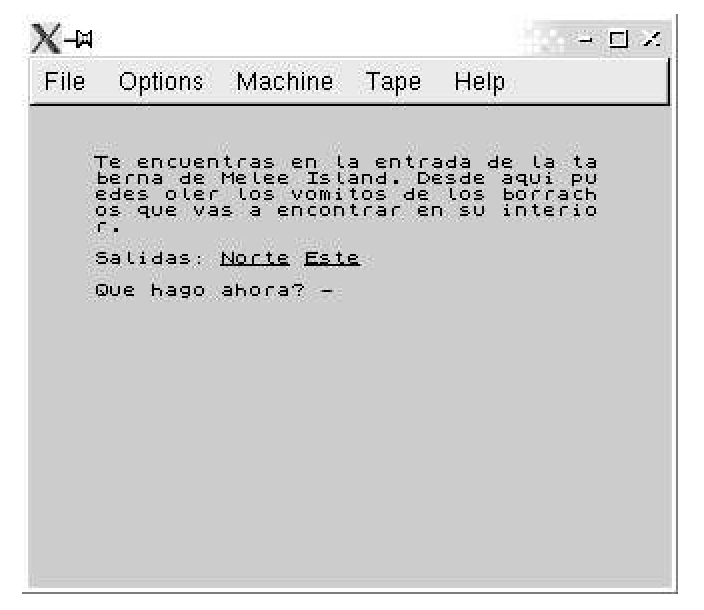
Las direcciones son escritas de forma especial. Además de un espacio en blanco al inicio de cada una, para separarlas en el caso de que se muestren varias por pantalla, existen unos carácteres raros al inicio (%c[4m) y al final (%c[24m). Esto es una particularidad de z88dk, y nos permite dar formato al texto. Para indicar que se va a formatear el texto, se introduce el carácter 27 en la cadena (mediante %c), y para el formato en sí mismo, se introduce [Xm, donde X es un código numérico que puede ser:

1 - texto en negrita
2 - texto delgado
4 - texto subrayado
7 - texto inverso

Si se utiliza 21, 22, 24 y 27 se desactiva el formato correspondiente. Así pues, al escribir una dirección, utilizamos el modo 4 para usar subrayado, y cuando terminamos de escribirla, utilizamos el modo 24 para desactivar el subrayado. Si no hiciéramos esto último,

cualquier cosa que escribiéramos a continuación aparecería subrayada. Si usamos [m (sin número entre [y m), se restaurarán los atributos originales del texto.

Si compilamos y ejecutamos, veremos nuestro primer resultado: se mostrará la descripción de la habitación inicial, y se nos mostrarán las direcciones de salida... y ahí acabará el programa, pues no hemos dicho que haga nada más.



Comienza la aventura

El intérprete de comandos

Para añadir interactividad, debemos introducir un intérprete de comandos, que ante los mandatos tecleados por el jugador, haga que pase algo en el juego. Este interprete de comandos consistirá en un bucle infinito en el que leeremos cadenas introducidas por el jugador, y según el valor de dichas cadenas se ejecutarán unas intrucciones u otras.

Para nuestro intérprete de comandos necesitaremos dos variables (que una vez más deben ser declaradas al inicio de la función main):

```
int final = 0;
char comando[250];
int i;
```

La primera es una variable que podrá tomar dos valores, 0 y 1. Cuando valga 1, querrá decir que la aventura habrá terminado y que el intérprete de comandos deberá dejar de ejecutarse. En el array de caracteres comando almacenaremos el comando de texto introducido por el jugador mediante el teclado. La variable i no es más que un contador que utilizaremos más adelante.

El intérprete de comandos se ejecutará mientras la variable final valga 0 (lo cual significa que la aventura no ha terminado), pidiendo un comando al usuario. Se comparará la cadena introducida por el usuario con un conjunto de cadenas o comandos que el interprete entenderá. Si la cadena se corresponde con alguno de los comandos entenidos por el programa, se desencadenarán las acciones pertinentes. En caso contrario, se mostrará un mensaje por defecto (del tipo de "No entiendo lo que dices").

Un ejemplo sencillo de intérprete de comandos que solo entiende los comandos "m" y "mirar" (ambos equivalentes) que borrarán la pantalla y mostrarán de nuevo al descripción de la habitación que nos encontramos se muestra a continuación, en el siguiente método main que deberá ser utilizado en lugar del ejemplo anterior:

```
void main(void)
{
    int habitacion = 0;
    int final = 0;
    char comando[250];
    int i;

    THabitacion habitaciones[6];
    inicializarHabitaciones(habitaciones);
    escribirDescripcion(habitaciones,habitacion);
    while (final == 0)
    {
        printf("Que hago ahora? - ");
        gets(comando);

    if (strcmp(comando,"m") == 0 || strcmp(comando,"mirar") == 0)
    {
        for (i=0;i<32;i++)
            printf("\n");

        escribirDescripcion(habitaciones,habitacion);
    }
    else
        printf("\n\nNo entiendo lo que dices\n\n");
   }
}
```

Este intérprete de comandos, que comienza con la sentencia while, se explica a continuación:

- La instrucción while (final == 0) hace que el código del intérprete de comandos se repita infinitamente, porque el valor de la variable final no va a cambiar.
- El primer printf indica al usuario que está esperando que introduzca un comando.
- La instrucción gets, al igual que el equivalente de la librería estándar de C del PC, espera a que el usuario introduzca el comando y lo almacena en la cadena comando, creada anteriormente.
- El if comprueba si el comando introducido es exactamente igual a alguno de los dos únicos comandos que entiende nuestra aplicación, m y mirar. Para ello se hace uso de strcmp, porporcionado una vez más por z88dk y una vez más de uso similar a la función homónima de la librería estándar de C de PC.
- Si el comando introducido no se corresponde con ninguno de los que entienda el intérprete, se ejecutará la instrucción tras el else, mostrando por pantalla un mensaje de error para que el jugador sepa que el programa no le ha entenido.
- Si el comando introducido es igual a m o mirar, lo que se hace es borrar la pantalla (escribiendo varios retornos de carro) y se llama a escribirDescripcion para que se muestre la información del a habitación actual. El método strcmp devuelve 0 si las dos cadenas pasadas como parámetro son iguales, y un valor distinto de cero en caso contrario.

Si compilamos el código y lo ejecutamos en el compilador, veremos que por fin ya podremos hacer algo con nuestro juego, aunque muy poco. Ya queda menos para que podamos movernos por nuestro mundo de aventura.

Moviéndonos por las habitaciones

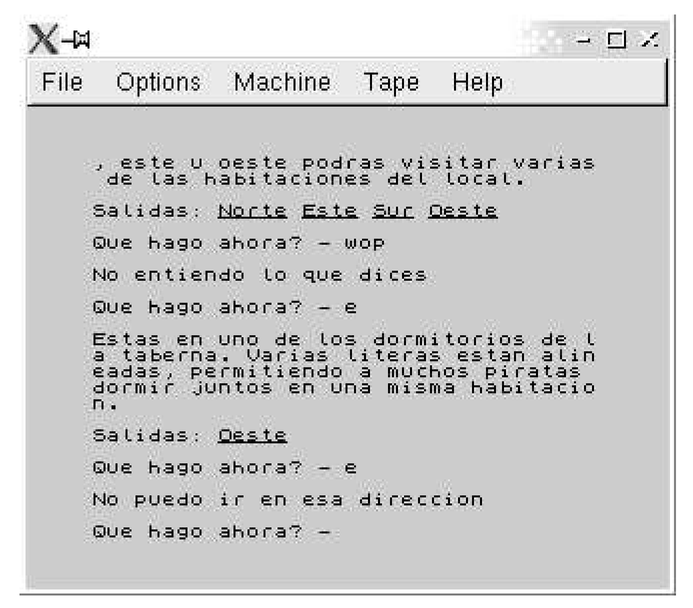
Por último explicaremos cómo hacer que el jugador pueda desplazarse por el mapeado de nuestra aventura conversacional. Deberemos hacer que nuestro intérprete de comandos entienda los comandos referentes al movimiento. Una forma de hacerlo es mediante el siguiente código:

```
while (final == 0)
{
    printf("Que hago ahora? - ");
    gets(comando);
    if (strcmp(comando,"m") == 0 || strcmp(comando,"mirar") == 0)
    {
        for (i=0;i<32;i++)
            printf("\n");
        escribirDescripcion(habitaciones,habitacion);
    }
    else if (strcmp(comando,"n") == 0 || strcmp(comando,"norte") == 0)
    {
        if (habitaciones[habitacion].direcciones[0] != 0)
        {
            habitacion = habitaciones[habitacion].direcciones[0] - 1;
            printf("\n\n");
            escribirDescripcion(habitaciones,habitacion);
        }
        else
            printf("\n\nNo puedo ir en esa direccion\n\n");

    }
    else if (strcmp(comando,"e") == 0 || strcmp(comando,"este") == 0)
    {
        if (habitaciones[habitacion].direcciones[1] != 0)
        {
            habitacion = habitaciones[habitacion].direcciones[1] - 1;
            printf("\n\n");
            escribirDescripcion(habitaciones,habitacion);
        }
        else
            printf("\n\nNo puedo ir en esa direccion\n\n");
    }
    else if (strcmp(comando,"s") == 0 || strcmp(comando,"sur") == 0)
    {
        if (habitaciones[habitacion].direcciones[2] != 0)
```

```
        {
            habitacion = habitaciones[habitacion].direcciones[2] - 1;
            printf("\n\n");
            escribirDescripcion(habitaciones,habitacion);
        }
        else
            printf("\n\nNo puedo ir en esa direccion\n\n");
    }
    else if (strcmp(comando,"o") == 0 || strcmp(comando,"oeste") == 0)
    {
        if (habitaciones[habitacion].direcciones[3] != 0)
        {
            habitacion = habitaciones[habitacion].direcciones[3] - 1;
            printf("\n\n");
                        escribirDescripcion(habitaciones,habitacion);
                }
                else
                        printf("\n\nNo puedo ir en esa direccion\n\n");
         }
         else
               printf("\n\nNo entiendo lo que dices\n\n");
    }
}
```

Lo único que se ha hecho ha sido introducir una sentencia if por cada comando que entiende el intérprete (después del else correspondiente al if anterior para evitar comparación con todos los comandos si ya se ha entenido qué es lo que el jugador quiere decir).



Moviéndonos por el mundo que hemos creado

Si el comando introducido por el jugador se corresponde con una dirección, se comprobará si se puede ir en esa dirección (examinando el valor de la posición adecuada en el array direcciones de la habitación actual), y en caso de que sea posible, se cambiará el valor de la variable habitacion (que marca la habitación en la que se encuentra el jugador actualmente) y se mostrará la descripción de dicha habitación. Si no es posible el movimiento en dicha dirección, se mostrará un mensaje adecuado.

Y ya está... si compilamos y ejecutamos el .tap en el emulador, veremos como ya podemos movernos por nuestro mundo conversacional.

¿Y ahora qué?

Hemos visto cómo utilizar un poco las funciones de entrada y salida y de cadenas de z88dk. Hemos comprobado que son iguales prácticamente a las del C estándar de los PCs actuales, excepto por la particularidad del formateo de texto visto al escribir las posibles direcciones por las que movernos (y alguna otra particularidad que todavía queda por describir).

A partir de ahora, crear la aventura conversacional será tan fácil como añadir nuevos comandos inteligibles por nuestro intérprete de comandos, mediante más sentencias if, y creando las estructuras de datos adecuadas para que eso sea posible. En próximas entregas crearemos objetos y haremos que el jugador pueda interactuar con ellos, al mismo tiempo que veremos algunas opciones más de formateo de texto.

LINKS

- Archivos fuente del ejemplo propuesto
  http://www.speccy.org/magazinezx/revistas/7/src/fuentes.tar.gz



SROMERO (NOP)

Entrevista a Pedro Gimeno

La década de los 90 supuso el declive comercial del Spectrum y la popularización del compatible IBM-PC. Fue entonces cuando muchos de nosotros pudimos seguir disfrutando de los juegos de nuestro pequeño amigo en la nueva plataforma de ocio y trabajo, gracias a un sorprendente programa publicado a finales de 1992 en la revista PCManía.

"Mi objetivo con el primer emulador era que mi Spectrum no se perdiera aunque se me estropeara."



"Spectrum.exe", el primer emulador de Spectrum español, trajo los ficheros en formato SP a nuestros discos duros, e inició el fenómeno de la emulación de Spectrum al alcance de todos los usuarios españoles. Este mes entrevistamos a su creador, Pedro Gimeno.

Cuéntanos cómo te introdujiste en el mundo del Spectrum. ¿Fue de la mano de los juegos, o te atrajeron sus posibilidades a nivel de máquina de cómputo?

Pues siendo el año 1981 y por tanto estando aún en sus albores la computación doméstica, no es de extrañar que lo que más me fascinaba fuera el poder de cómputo y la inmensidad de posibilidades que parecía ofrecer. Pero claro, no hay que olvidar que por aquellas fechas yo tenía trece años, por tanto la idea de aprovechar ese potencial para jugar era casi natural a esa edad.

¿Cuál ha sido tu evolución dentro de los diferentes modelos de ordenadores antes y después del Spectrum?

Empecé con un ZX-81 que me prestaron durante bastante tiempo. Con él aprendí ya bastante sobre el BASIC de Sinclair, pero no tardé en darme cuenta de que los programas más interesantes estaban hechos en código máquina, y ¡oh, bendición del cielo!, el apéndice A del manual incluía una lista de las instrucciones del Z80. Mis primeros programas en código máquina los escribía tecleando directamente una línea REM con las letras cuyo códigos eran los de las instrucciones que necesitaba. He eludido mencionar "códigos ASCII" porque el ZX-81 no usaba el estándar ASCII. En el ZX-81 había sólo unos pocos códigos que no tenían un carácter equivalente, y esos los cambiaba mediante POKEs a posteriori. Aún recuerdo la famosa secuencia "ERND" por la que empezaban muchos programas; esto equivalía a "ld hl,(DFILE)", es decir, cargar HL con la dirección de la pantalla. También recuerdo que el programa se ejecutaba con RAND USR 16514, que era la dirección del carácter que seguía al REM.

Cuando vi las especificaciones del ZX-Spectrum me quedé maravillado (no es de extrañar viendo las del pobre ZX-81). Mi primer contacto con un Spectrum fue en casa de un compañero de colegio; por desgracia no hubo tiempo para mucho, apenas para probar el PLOT y comprobar la alta(?) resolución (recordemos que la resolución del PLOT del ZX-81 era de 64x44 a base de caracteres semigráficos).

Me costó mucho convencer a mis padres para tener uno, pero finalmente accedieron. No estoy seguro del año; es muy probable que fuera en 1983. En cuanto lo tuve lo primero que hice fue averiguar cómo se hacía en el Spectrum todo lo que estaba acostumbrado a hacer en el 81: dónde estaba la memoria de pantalla, cómo estaba organizada (menudo chasco cuando vi la extraña estructura), cómo escribir código en REMs y todas esas cosas. No tardé mucho en darme cuenta de que el sistema de los REMs no era el más indicado;

los códigos de control introducidos con el cambio a ASCII eran un obstáculo muy fuerte, porque había bastantes instrucciones importantes en el rango 0-31; notablemente el 17, que era un código de control que además producía errores al listar. Tuve que mudarme a emplear la nueva instrucción DATA y escribir con números decimales en vez de caracteres.

Luego por supuesto apareció el típico vecino que tenía un amigo que tenía muchos juegos. De entre los primeros que me llegaron recuerdo con especial cariño el Scuba Dive y, sobre todo, cómo no, el Manic Miner. Cabe decir que mis padres no aprobaban la compra de juegos y yo no tenía dinero, así que la opción de tener originales no estaba entre las posibles.

El siguiente ordenador que tuve fue un PC, un 286 a 10 MHz con tarjeta EGA y 70 Mb, que aunque en realidad era de mi padre yo era el principal usuario. Para entonces ya había utilizado muchos PCs y conocía bastante su estructura. Era de procedencia francesa; la marca se llamaba Goupil. Entre la documentación se encontraban manuales técnicos del hardware y el código fuente de la ROM; esto me ayudó bastante a profundizar en el funcionamiento interno de un PC con todos sus periféricos, sobre todo la tarjeta gráfica.

Más adelante comencé a trabajar y a tener autonomía económica. El primer ordenador que compré fue un Sinclair QL, aunque ya se había pasado un poco la época de su máximo esplendor. Era técnicamente bastante atractivo, salvo por tres inconvenientes básicos: era algo lento (basado en un 68008), tenía poco software y funcionaba mediante los odiosos Microdrives. Esto último me llevó al extremo de tener que devolverlo, ya que desde que lo compré estaba constantemente teniendo que llevarlo a reparar porque el cabezal de lectura se desplazaba y dejaba de leer. Supongo que estaba defectuoso. El software que incluía, sin embargo, era muy bueno y guardo buenos recuerdos de él.

También compré un Atari 1040 ST, pero el uso que le di fue casi exclusivamente dedicado a la música por MIDI. No profundicé en sus posibilidades computacionales.

A partir de ahí todo lo que he tenido han sido PC's, excepto por un Commodore 128 que me regalaron porque querían a deshacerse de él, pero que no llegué a usar (no tenía software). Más adelante también conseguí un ZX-81 en buen estado, pero tampoco he llegado más allá de comprobar que funcionaba. También un Inves Spectrum sin fuente de alimentación, que no sé si funciona.

Y de todos estos equipos, ¿cuál es tu preferido o de cuál guardas mejores recuerdos?

El Spectrum, por supuesto. ¿Acaso lo dudabas?

Debo decir de todas formas que cuando preparé mi segundo emulador en el Pentium 200, me maravillé de que tuviera la potencia suficiente como para realizar una emulación completa en tiempo real, ciclo a ciclo. Viendo que iba sobrado de velocidad no me preocupaba mucho añadir aspectos más finos de la emulación, porque sabía que los soportaría. Esto me permitió decidirme a incorporar emulación de 128K, opciones de sonido y otros refinamientos.

¿Qué ocurrió con este tu "segundo emulador"?

La idea partió de la documentación del x128 de Joe McKay, creo que en la versión 0.5. Este emulador incorporaba una opción llamada "ULA Delay" en un intento de que se vieran demos como la Overscan de Busy Soft. El autor se quejaba de que el número de ciclos parecía ser un número fraccionario, cosa que no alcanzaba a entender. Al leerlo sospeché que el problema consistía en que el retraso no era homogéneo, sino que dependía de algún factor como a qué dirección se accedía en qué momento. Así que me lie la manta a la cabeza y me puse a investigar sobre la "contended memory" hasta que tuve todos los detalles bien perfilados.

Sólo faltaba aplicar mis descubrimientos. Mi objetivo con el primer emulador era que el Spectrum no se perdiera aunque se me estropeara, pero sabía que la emulación era imperfecta. Pero tras mis averiguaciones tenía, por primera vez, la oportunidad de realizar una emulación prácticamente perfecta. Esto significaba, por supuesto, renunciar a que se usara en máquinas de potencia modesta, pero el precio a pagar era pequeño en comparación con la ganancia. Así que de nuevo me puse manos a la obra empezando un nuevo emulador desde cero, pretendiendo incorporarle todos los refinamientos de emulación de que fuera capaz.

Para entonces ya empezaban a existir unos cuantos "DOS extenders" y el NASM acababa de salir. Decidí pues entrar en el maravilloso mundo de los 32 bits y el DPMI, esta vez usando íntegramente herramientas FOSS. Por el camino me enfrenté a algunos errores en el NASM que localicé y reparé (y de los cuales por supuesto informé a su autor, que para entonces aún era Simon Tatham, a quien también debemos hoy el PuTTY).

El primer extensor que utilicé creo que era el PMODE/W. Mientras programaba el emulador y le iba añadiendo cosas me amenazaba cada vez más la idea de que el resultado no funcionara con la suficiente velocidad ni siquiera en un Pentium 200. Puedes imaginar la alegría que me llevé al hacer las primeras pruebas de temporización, sólo comparable a aquella primera vez en la que vi el Manic Miner aparecer en la pantalla.

Más adelante, ante la perspectiva de portarlo a Linux, renuncié a usar únicamente ensamblador y decidí escribir en C la interfaz con el sistema operativo, con lo cual pasé a usar DJGPP y su extensor. Por entonces ya había pasado por varios extensores y linkers, incluyendo el linker de RDOFF del NASM; probablemente sea de las pocas personas del mundo que le han sacado utilidad, en parte gracias al WDOSX que lo soportaba.

Este emulador fue el primero en el que se vio la demo Overscan correctamente. Pero el tiempo que podía dedicarle no era mucho, y el retraso en el lanzamiento me hizo pensar que tal vez era mejor publicar la información que había recopilado con la esperanza de que otros la aprovecharan. Y así fue, pero sólo el FAQ de comp.sys.sinclair me otorgó

crédito por ello. Como resultado mi nuevo emulador cayó básicamente en el olvido. Con la aparición del RealSpectrum vi satisfechas mis propias necesidades en cuanto a emulación y decidí no proseguir con el desarrollo.

En tu época en el mundillo del Spectrum, ¿llegaste a tomar contacto con algún grupo de programación de compañías como Dinamic, Made in Spain, Zigurat u Opera?

No, en absoluto. En el Spectrum no destacaba por ningún programa. Lo que más hacía era jugar, buscar pokes cuando el juego era demasiado difícil para mí, lo cual ocurría con frecuencia, y desproteger programas para copiarlos (he de decir en mi descarga que no tenía dinero y mis padres no apoyaban que jugara). Para mí desproteger era un desafío interesante; llegué a poder con algunas protecciones complicadas como por ejemplo el famoso Speedlock.

Sin embargo, más tarde, como resultado de escribir el emulador, Víctor Ruiz se puso en contacto conmigo para pedirme una adaptación del juego After The War al PC usando el emulador mismo. Accedí y el juego fue publicado, aunque sin control de velocidad de ninguna clase. Los únicos cambios fueron: una optimización de la velocidad para la instrucción LDIR hecha a medida del programa, y una adaptación para evitar que necesitara la ROM del Spectrum, ya que según me aseguró el juego no requería de la misma, y por motivos de copyright no querían incluirla.

¿Llegaste a programar algún programa o juego complejo o extenso para Spectrum?

Realmente no. Los programas más complejos que recuerdo fueron uno para probar autómatas celulares lineales y uno que hacía de buzón de correo para radio que en realidad nunca se llegó a poner en marcha.

También hice un programa para mostrar el conjunto de Mandelbrot, pero usando coma flotante lo cual lo hacía tremendamente lento incluso siendo compilado. Llegué a obtener una imagen decente en blanco y negro del conjunto cuando aún no conocía (¿no existía?) el ahora famoso Fractint. Era el año 1985; A.K.Dewdeney acababa de publicar su famoso artículo sobre este fractal en la revista Investigación y Ciencia, a la que mi padre estaba suscrito. Por entonces no conocía las técnicas de cálculo en coma fija mediante enteros; si lo hubiera programado ahora seguramente me hubiera salido mucho más rápido.

Pasando a hablar de tu emulador Spectrum, ¿cuándo y cómo se te ocurrió la idea de programar un emulador de Spectrum para PC?

Bueno, para cuando tuvimos el Goupil las casas que reparaban Spectrums empezaban a escasear y yo temía que llegara el momento en el que se me estropeara y no pudiera arreglarlo. Quería encontrar una forma de alargarle la vida y al mismo tiempo poder usar un teclado en condiciones. La idea de emularlo surgió casi sola a partir de esas premisas. Además corría la voz de que existía un emulador para Amiga, lo cual resultó ser verdad. No recuerdo de quién era, puede que fuera el de Peter McGavin.

Sabía que para emular un ordenador de 3.5 MHz con uno de 10 necesitaría código lo más optimizado posible y que aún así iría lento, lo cual más tarde se demostró cierto. No era posible escribir un emulador tal que por cada tres instrucciones de PC se consiguiera una de Spectrum. He de hacer notar que la técnica de la compilación Just In Time no se me ocurrió ni era ampliamente conocida. Sí que es cierto que el emulador de Amiga la empleaba, pero no tuve el menor contacto con él ni conocimiento de sus técnicas.

Empecé a escribirlo en Pascal pero renuncié enseguida, dándome cuenta de que en ensamblador sería mucho más sencillo de lo que me imaginé al principio, incluso más que en Pascal. Los manejos de memoria serían mucho más simples, y con una tabla de saltos paliaría bastante el problema de la velocidad. Además con los flags la CPU me ahorraría un montón de trabajo, y podría sacar ventaja del hecho de que el x86 tiene los flags en las mismas posiciones de bits que el Z80. Así que me lié la manta a la cabeza y me puse manos a la obra con un emulador en ensamblador.

Cuéntanos algo sobre el proceso de desarrollo de tu emulador. ¿Qué herramientas de desarrollo usabas? ¿De dónde obtuviste toda la información técnica necesaria para la emulación de ciertas instrucciones indocumentadas, de los timings de las mismas, etc.?

Mi editor favorito siempre ha sido uno llamado SEE que me presentó mi amigo Javier Vázquez. Con él editaba los .asm, y los ensamblaba con el TASM. Dada la extensión tuve que organizarlo por módulos (ficheros .obj que luego enlazabacon el TLINK). Aprendí a escribir Makefiles para no tener que compilar todo el proyecto cada vez que hacía un cambio.

Como información sobre el Z80 usé las microfichas de código máquina de Microhobby, por la comodidad del formato sobre todo. De la temporización no podía ocuparme dada la poca velocidad que esperaba obtener y de hecho obtuve. Además de las microfichas consultaba un libro sobre el Z80 de SGS proporcionado por mi amigo Diego Hernández.

El manual técnico del Goupil me fue extremadamente útil para recabar información sobre la EGA y sus detalles. Por desgracia, los modos estándar de la EGA eran muy lentos para escribir en ellos; había que escribir por planos, desglosando cada color en cuatro bytes distintos. El primer emulador que tuve a punto, el EGASPEC, tenía un cuello de botella tremendo en el vídeo.

Conocía un juego que aprovechaba una característica extraña de la CGA para implementar un modo de vídeo de 160x100 a 16 colores. Lo hacía mediante un modo de texto, alterando la altura de los caracteres. En una conversación que mantuve con Alejandro Valero sobre el problema de la velocidad gráfica, durante una tormenta de ideas se nos ocurrió usar una técnica parecida a la de ese juego, utilizando un modo de texto, y redefinir los caracteres (en la CGA no era posible, pero en la VGA sí). Así le dimos forma al que sería el modo definitivo: caracteres de 8x1 con la misma forma que los bytes de Spectrum en la memoria de pantalla. Cuando

implementé ese modo, la mejora de la velocidad gráfica fue realmente espectacular.

¿Quieres decir que en lugar de utilizar píxeles en un modo gráfico, tu emulador utiliza CARACTERES DE TEXTO redefinidos para así dibujar múltiples píxeles de una sola vez?

Sí, en efecto. No sólo eso, es que además separa caracteres y atributos de la misma forma que en el Spectrum, y por tanto cada byte de memoria de pantalla del Spectrum equivale a un solo byte de memoria de pantalla del PC para la zona de píxeles. La información de atributos implica un poco más de trabajo, pero muy poco más. Sólo hay que escribir el byte ocho veces y añadir el procesado de flash; con eso queda completo. La ventaja que le da al emulador respecto a los modos gráficos convencionales es evidente.

Tu emulador fue incluído en uno de los discos de la revista PCManía de Diciembre de 1992. ¿Cómo surgió la posibilidad de que fuera publicado en PCManía?

Me llamaron ellos para pedirme permiso para publicarlo. Me pidieron incluso que se lo diera por escrito, y así lo hice. Me prometieron una suscripción a cambio, pero nunca cumplieron su palabra.

Después de trabajar en el emulador, ¿te has vuelto a involucrar en algún proyecto de software, ya sea para Spectrum o para otras arquitecturas?

Sí, he dedicado parte de mi tiempo a dos proyectos de software libre: GMP y GIMP. En el primero he implementado una versión del generador de números pseudoaleatorios Mersenne Twister; en el segundo he colaborado arreglando algunos bugs.

Ya por último, ¿a qué se dedica Pedro Gimeno actualmente? ¿Cuáles son sus hobbies y sus ocupaciones?

Mi hobby actual se centra en seguir colaborando para el GIMP. Mi ocupación es un trabajo que consiste básicamente en introducir datos. Sí, yo también me pregunto qué hace un programador haciendo de usuario, pero al menos mi trabajo me permite vivir.

Muchas gracias por tu atención, Pedro. Desde MagazineZX queremos agradecerte que nos hayas dedicado tiempo para esta entrevista y aprovechamos para felicitarte por lo que supuso para muchos de nosotros la creación de un emulador como "Spectrum", que nos permitió continuar viviendo durante mucho tiempo la magia de nuestro querido Sinclair en una nueva arquitectura como era el PC. No sé si eres consciente de que muchos de nosotros poblamos nuestros discos duros con ficheros .SP desde finales de 1992...

Gracias a vosotros. Un saludo a todos y buena suerte con la revista.



Esta fue la pantalla que introdujo a muchos en el mundo de la emulacion.



SROMERO (NoP)

En esta entrega, SROMERO (NoP) nos da su opinión acerca del mercado de compra-venta de productos Sinclair.

Vendo, compro y cambio (anuncios curiosos)

Los ordenadores y consolas retro tienen un lugar destacado en las operaciones de compra-venta de todo *freaky* que se precie de serlo. El coleccionismo de todo tipo de consolas, juegos portátiles y microordenadores es la diversión de muchos de nosotros (así como la maldición de muchas madres y/o esposas). En general cualquier cosa que lleve pilas y/o se enchufe a la red eléctrica es susceptible de ser coleccionada.

Cuando un producto tiene mucha demanda y poca oferta siempre aparece quien desea sacar provecho de esta situación, y los precios suben. En el caso del Spectrum, muchos de nosotros hojeábamos la sección de Compra-Venta de Microhobby. Lo triste del asunto es que más que hacerlo para comprar productos, lo hacíamos para reirnos con algunos de los anuncios. Gente queriendo cambiar su ZX81 por un Spectrum 48K, ventas de segunda mano por precios superiores al producto nuevo, etc.

Hoy en día la cosa no ha cambiado mucho, y seguimos pudiendo encontrar en Internet ofertas que harían palidecer al mejor de los canales de compra-venta de aparatos abdominales definitivos. Basta con acudir a eBay o a páginas web de particulares para ver precios desorbitados por material hardware y software de nuestro querido Spectrum. Como diría nuestro amigo S.T.A.R., las cosas valen lo que uno esté dispuesto a pagar por ellas... y esto es algo de lo que se aprovechan determinadas personas, nos gustaría pensar que por desconocimiento.

Quiero destacar que esto no es ningún alegato contra los coleccionistas. Es más, yo mismo soy coleccionista (como dije anteriormente, de todo lo que se enchufe o lleve pilas. Lo que quiero mostrar en este comentario es el nivel al que se llega en la compraventa de material, tanto desde el punto de vista del vendedor, como del comprador.

Para ilustrar esto hemos recopilado una serie de anuncios de ejemplo tomados de diferentes lugares de la geografía Internetera. Gracias a Horace por permitirnos reproducir aquí fragmentos de anuncio de su web (ESpectrum). Si algún anunciante se reconoce en los siguientes párrafos, espero que no se sienta ofendido.

Los anuncios

- Vendo Spectrum 128K, con juegos, joystick y en perfecto estado. Todos los juegos que en los años 80 existieron. Espero ofertas. Valladolid

Wow. Un Spectrum con 10.000 juegos... No quiero ni pensar qué precio pedirá por él. En cualquier caso, algo de lo que hay que hablar es del precio de las cintas de juegos (es más complicado encontrar discos de 3" originales). Hace unos años, era fácil encontrar cintas de Spectrum en los rastros de las ciudades a 50 pesetas la cinta (por el método universal de "2 cintas, 20 duros"). Con la llegada del euro, a veces se siguen pudiendo encontrar cintas, a 3 cintas 1 euro, o 2 cintas un euro. Existen vendedores del rastro que no saben ni qué es lo que venden, y te dicen "1 euro éstas, y 3 euros ésas. Son videojuegos para ordenadores, nuevos valen 5.000 pesetas". Por supuesto, si les dices "Perdone, pero eso es imposible, estos ordenadores hace más de 10 años que no se venden" te responden con "Pues más motivo aún, son difíciles de encontrar". Vivir para ver. En general, los precios de las cintas en particulares oscilan entre los 50 céntimos y los 3 euros, según el tipo de caja (no es lo mismo caja de lujo, que cajita de plástico) y según el título. Está claro que no es lo mismo comprar Manic Miner que Kung Fu Master... (como tampoco es lo mismo una cinta usada, que una por desprecintar).

- Caja vacía de un Spectrum 48k. 30 euros.

Bueno, como podéis ver, hemos llegado a un punto en que los embalajes por sí sólos casi tienen un precio superior al del propio aparato. El negocio del siglo es poner 3 anuncios: uno en que vendemos el Spectrum completo con caja, y luego uno vendiendo la caja y otro vendiendo el Spectrum (por separado). A ver quién pica antes... Resulta curioso ver cómo un Spectrum casi dobla su valor sólo añadiéndole la palabra "BOXED" (empaquetado en su caja original), sobre todo si está en buen estado. Hay quien le echa suficiente morro, y anuncia su Spectrum "CORCHED" (en su corcho original). Si algún día tengo que vender un Spectrum, me aseguraré de que sea "ZAPATED" (tengo varias cajas vacías de zapatos para envíos que dan el pego).

- Vendo spectrum 48k teclado de goma por menos de 300 euros y juegos originales de los primeros que salieron como cowboys...

¡Corred! ¡Últimas unidades! ¡Y por menos de 300 euros! ¡Yahoooooo! (esto último es por lo de Cowboys).

- Vendo Spectrum 48 k, con transformador, cables y manuales en su caja original. Incluyo Impresora Térmica Alphacom 32, con 3 rollos de papel térmico, Interface y Joystick. Tres libros. Programa para hacer quinielas. Compilador de Pascal y varios Juegos. Precio 210 euros.

Bueno, 210 euros son nada más y nada menos que 35.000 de las antiguas pesetas. Está claro que la impresora térmica y los libros son un buen añadido. El compilador de Pascal también tendrá su precio, pero... ¿realmente 35.000 pts es un precio adecuado?

- Vendo revistas Microhobby a 5 euros el numero.

Ahh... las revistas Microhobby... ¡quién pudiera tener la colección completa! Hubo una época en que se podían comprar Microhobbies atrasadas (de segunda mano) por 100 pesetas (0.60 euros) la unidad. Han ido subiendo de precio paulatinamente (es normal, es un artículo de primera necesidad, como el pan y la gasolina), pasando por 150 pts, 200 pts, 250 pts, y 300 pts por número. Con la llegada del euro (y el redondeo) pasamos a 2 euros (333 pts) y 3 euros (500 pts) por número. Y ya el no va más son los 5 euros por número; vamos que si la colección de Microhobby son 224 ejemplares, nosotros tenemos que pasar de las 186.000 pts (1120 euros) para poder comprarla. Está claro que el vendedor avezado está jugando con el "comprador de números sueltos", es decir: alguien que se gastará unos "míseros" 50 euros con el fin de tener las 10 revistas que le faltan para llenar huecos en su colección.

- Compraría ZX 81 que funcione

Je. Vaya preparando la VISA, o acercándose a su banco más cercano, porque la cosa puede salir cara... Comprar un ZX80 es poco menos que imposible (obviamente, seguro que se puede hacer a golpe de talonario, pero está muy lejos de los bolsillos del estudiante *freakie* medio). El ZX81 es más accesible, pero sigue siendo caro (caro entre comillas, porque si realmente lo quieres, pagarás cualquier cantidad, sin pasarse, por disfrutar de él). Sin duda es mucho más asequible un Spectrum (más concretamente, el más barato es el modelo +2A).

- Vendo ZX spectrum +2a 128k con pistola y juegos para esa pistola fuente alimentacion nuevo y regalo con el lote monton revistas y juegos y un digitalizador imagen y tambien ademas doy 2 ordenadores con su embalaje nuevos sin extrenar que te serviran por si te se rompe algun dia alguna pieza incluso las fuentes de alimentaccion hay bastantes juegos originales el lote lo vendo 30.000 pts.

¿Tres ordenadores por 30.000 pts? ¿Con su caja? ¿Con un digitalizador de imagen? ¿Esto es caro, o barato? Y de "regalo" un montón de revistas y juegos. Lo que parece seguro es que el transporte de este lote lo tendrá que realizar una grúa...

- Vendo cintas de spectrum repletas de juegos. (unas 400 mas o menos con mas de 3000 juegos) 100 euros.

400 cintas, con 3000 juegos, son 7 juegos por cinta. O sea, que original, ni una. Pero es un chollo, porque las vende por sólo 100 euros... ¡Sólo falta que encima sean TDK, y de 90 minutos! En fin, en este mundillo de la compraventa, hay gente que todavía no se ha dado cuenta de que el 99% de cintas grabadas (no originales) no tienen el más mínimo valor: en WOS, El Trastero y SPA2 podemos encontrar prácticamente todo el software de Spectrum listo para regrabarlo en cinta. La única utilidad de ciertas cintas grabadas es encontrar versiones en cinta de juegos de los que no hay versión digital, programas realizados por el dueño de la máquina, o cracks caseros para los juegos que en ellas se grabaron.

Y, por último, mi favorita:

- ESTOY BUSCANDO UN SPECTRUM 128 DE LOS ANTIGUOS.

Recientes excavaciones en Egipto demuestran que Akenatón tenía un 128K. ¿Sería ese lo suficientemente antiguo?

Para terminar

Querría concluir esta sección de opinión con un consejo para compradores y vendedores. A los primeros les aconsejo que mediten sus decisiones antes de comprar: no se debe comprar a cualquier precio. No hay que dejarse llevar por el "date prisa que ya tengo varias ofertas" ni el "Debe de ser uno de los pocos que quedan". Hay que negociar siempre con el vendedor, e intentar llegar a un precio justo. El objetivo de todo esto es rebajar la inflación de precios. Si alguien oferta un +2A por 90 euros y lo vende, el siguiente que coja lo ofertará por el mismo precio o más, y nadie (salvo el comprador) podrá detener la escalada de precios. Siendo que hay +2A desde 30 euros, lo mejor que puede hacer el comprador es comparar y negociar.

Para los vendedores el consejo es sencillo pero moralmente complejo: por favor, maximizad vuestro beneficio pero minimizad el precio. Es totalmente lícito vender un producto al máximo precio posible y sacar todo el beneficio que se pueda pero, personalmente, me parece más bonito vender algo a una persona que realmente aproveche tu producto y, además, hacerlo a un precio justo. Sabemos que un +3 se puede vender a 120 euros, pero creo que es mejor venderlo a 100 euros a una persona que lo va a usar para programar en él nuevos juegos, que venderlo por 120 a alguien que lo quiere para hacerle una foto para su Web y meterlo en una vitrina. Es importante ajustar los precios para que todos salgan ganando. Si un +2A nos sale gratis (un regalo), ¿no sería mejor venderlo por 35 euros (y que todos puedan optar a él) que por 50 (y que sólo algunos puedan)?

Todavía me sorprendo cuando veo gente que regala (o vende a bajo precio) Spectrums, y que busca sobre todo el interés general. Es gracias a ellos que este mundillo sigue en marcha.

LINKS

- ESpectrum (la página de Horace): http://www.speccy.org/espectrum/
- World Of Spectrum: http://www.worldofspectrum.org/
- El Trastero del Spectrum: http://www.speccy.org/trastero/
- SPA2: http://www.speccy.org/spa2/